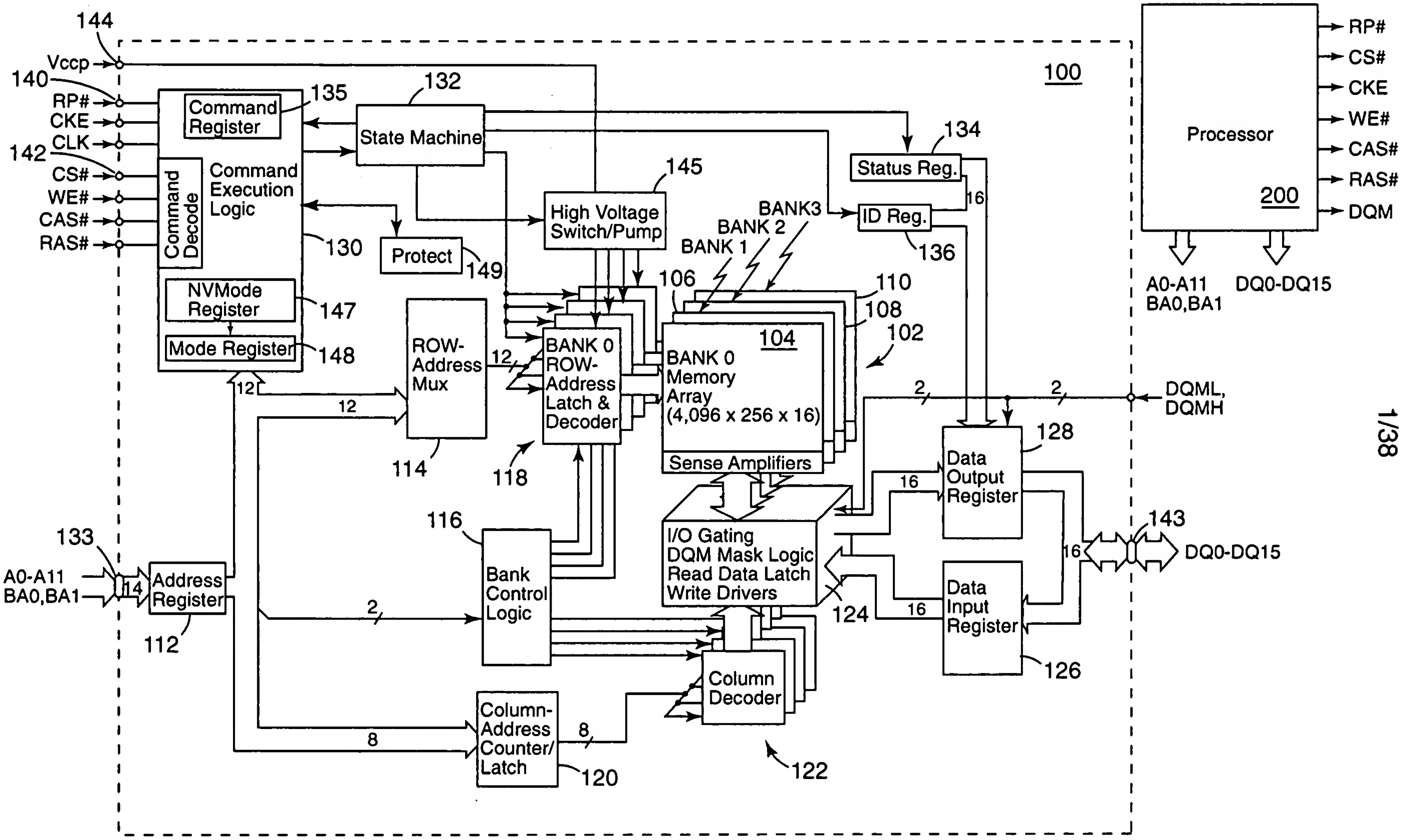

144
Vccp
140
RP#
CKE
CLK
142
CS#
WE#
CAS#
RAS#
Command Register
135
132
State Machine
Command Decode
Command Execution Logic
130
Protect
149
NVMode Register
147
Mode Register
148
12
12
ROW-Address Mux
114
High Voltage Switch/Pump
145
BANK3
BANK 2
BANK 1
106
BANK 0 ROW-Address Latch & Decoder
118
12
104
BANK 0 Memory Array (4,096 x 256 x 16)
Sense Amplifiers
110
108
102
Status Reg.
134
16
ID Reg.
136
100
Processor
200
RP#
CS#
CKE
WE#
CAS#
RAS#
DQM
A0-A11 BA0,BA1
DQ0-DQ15
DQML, DQMH
2
2
Data Output Register
128
16
16
143
DQ0-DQ15
Data Input Register
126
16
I/O Gating DQM Mask Logic Read Data Latch Write Drivers
124
133
A0-A11 BA0,BA1
14
Address Register
112
2
Bank Control Logic
116
8
Column-Address Counter/ Latch
120
8
Column Decoder
122

Fig. 1A

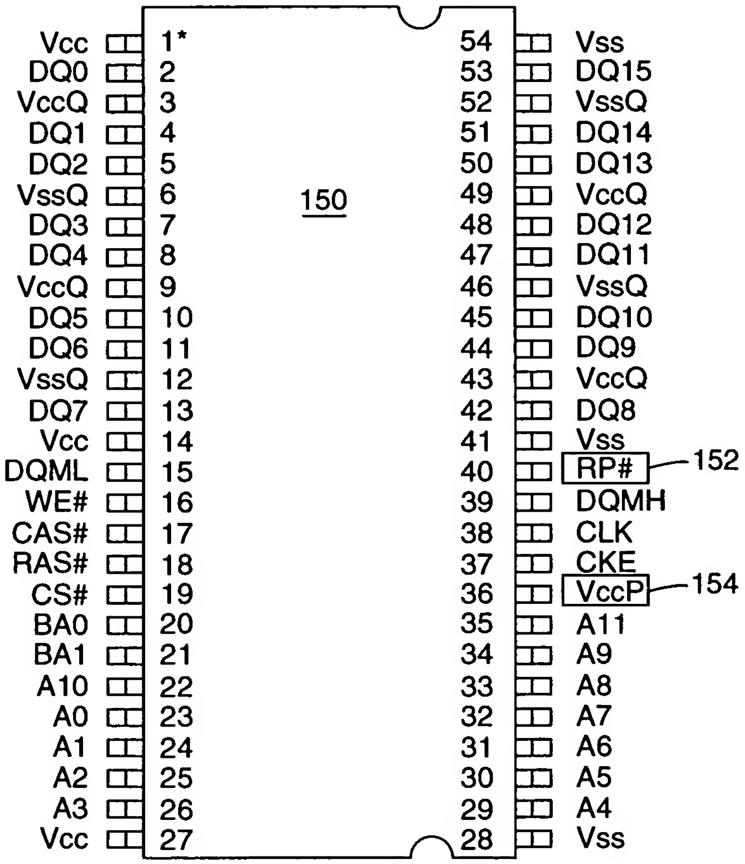
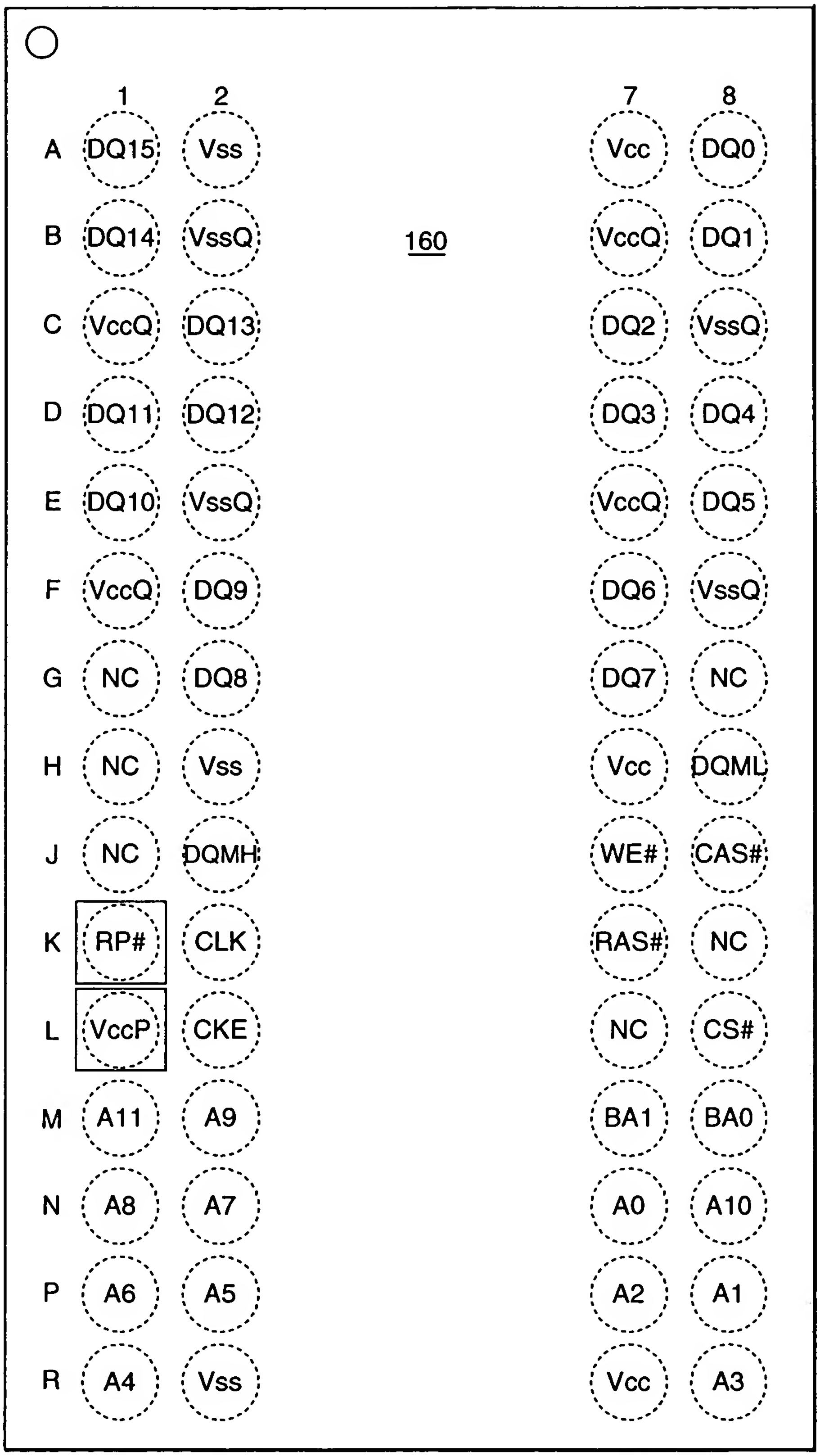

150
Vcc 1*
DQ0 2
VccQ 3
DQ1 4
DQ2 5
VssQ 6
DQ3 7
DQ4 8
VccQ 9
DQ5 10
DQ6 11
VssQ 12
DQ7 13
Vcc 14
DQML 15
WE# 16
CAS# 17
RAS# 18
CS# 19
BA0 20
BA1 21
A10 22
A0 23
A1 24
A2 25
A3 26
Vcc 27
54 Vss
53 DQ15
52 VssQ
51 DQ14
50 DQ13
49 VccQ
48 DQ12
47 DQ11
46 VssQ
45 DQ10
44 DQ9
43 VccQ
42 DQ8
41 Vss
40 RP#
152
39 DQMH
38 CLK
37 CKE
36 VccP
154
35 A11
34 A9
33 A8
32 A7
31 A6
30 A5
29 A4
28 Vss

Fig. 1B



	1	2	7	8
A	DQ15	Vss	Vcc	DQ0
B	DQ14	VssQ	VccQ	DQ1
C	VccQ	DQ13	DQ2	VssQ
D	DQ11	DQ12	DQ3	DQ4
E	DQ10	VssQ	VccQ	DQ5
F	VccQ	DQ9	DQ6	VssQ
G	NC	DQ8	DQ7	NC
H	NC	Vss	Vcc	DQML
J	NC	DQMH	WE#	CAS#
K	RP#	CLK	RAS#	NC
L	VccP	CKE	NC	CS#
M	A11	A9	BA1	BA0
N	A8	A7	A0	A10
P	A6	A5	A2	A1
R	A4	Vss	Vcc	A3

Fig. 1C

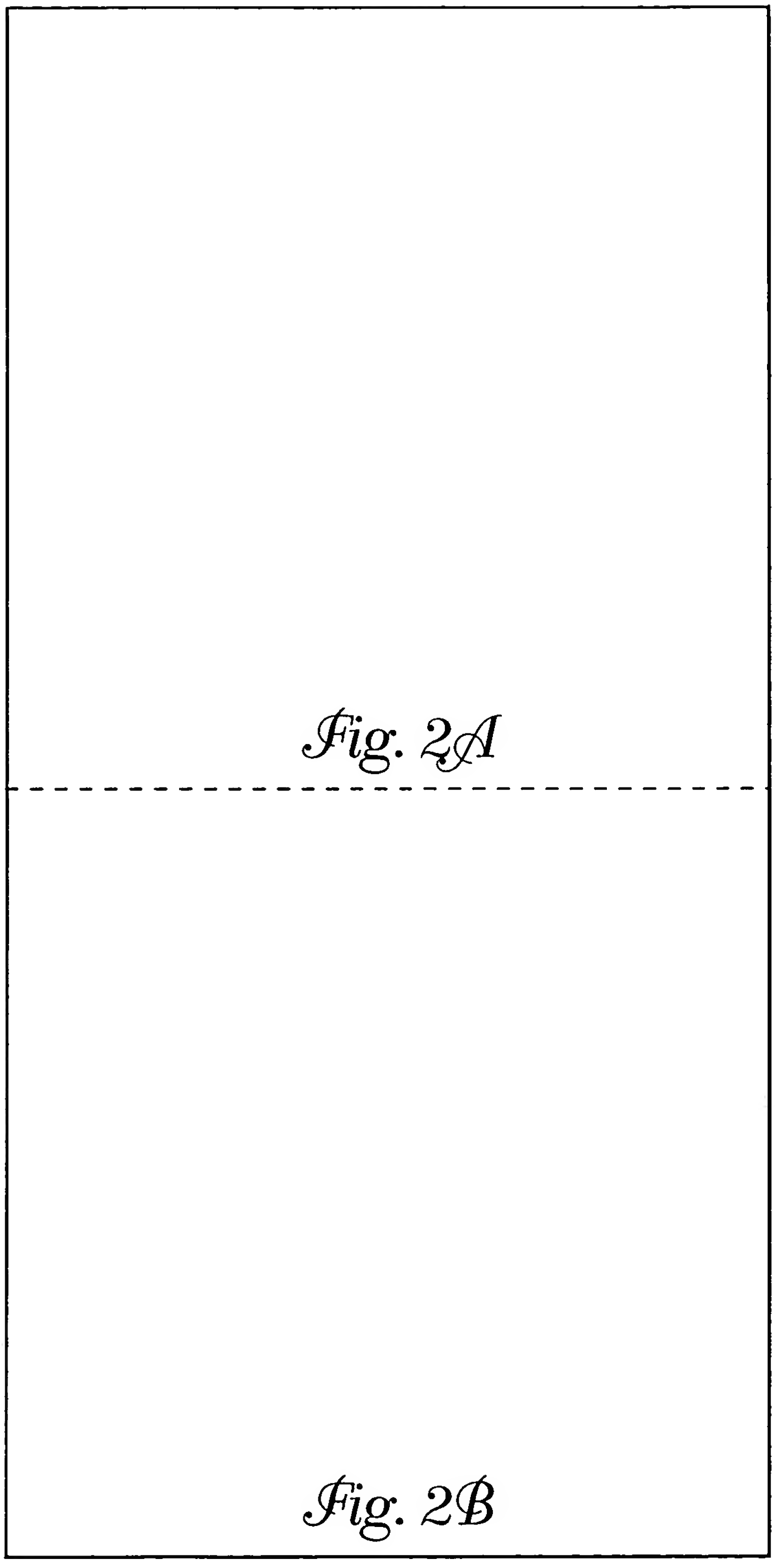
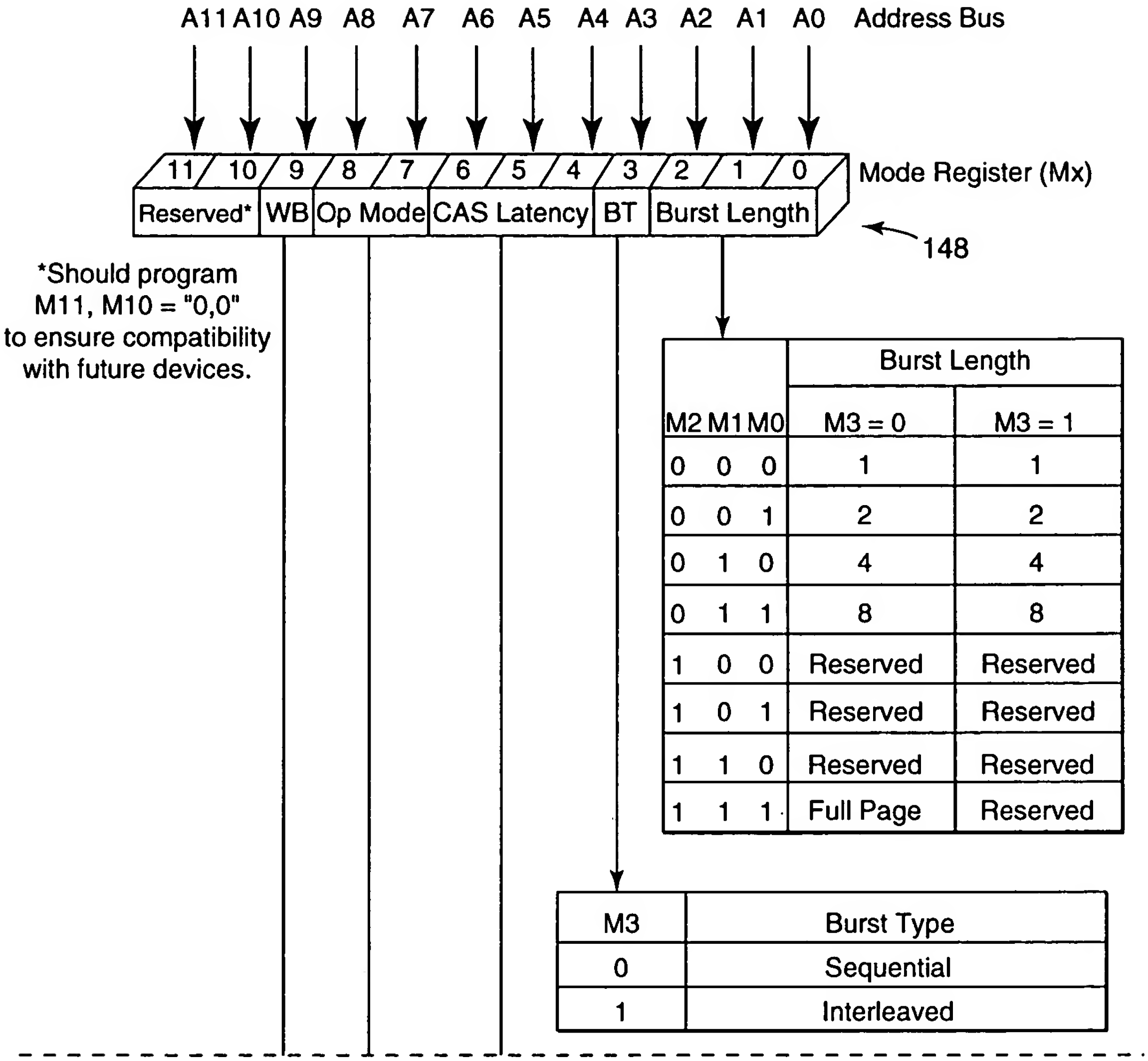


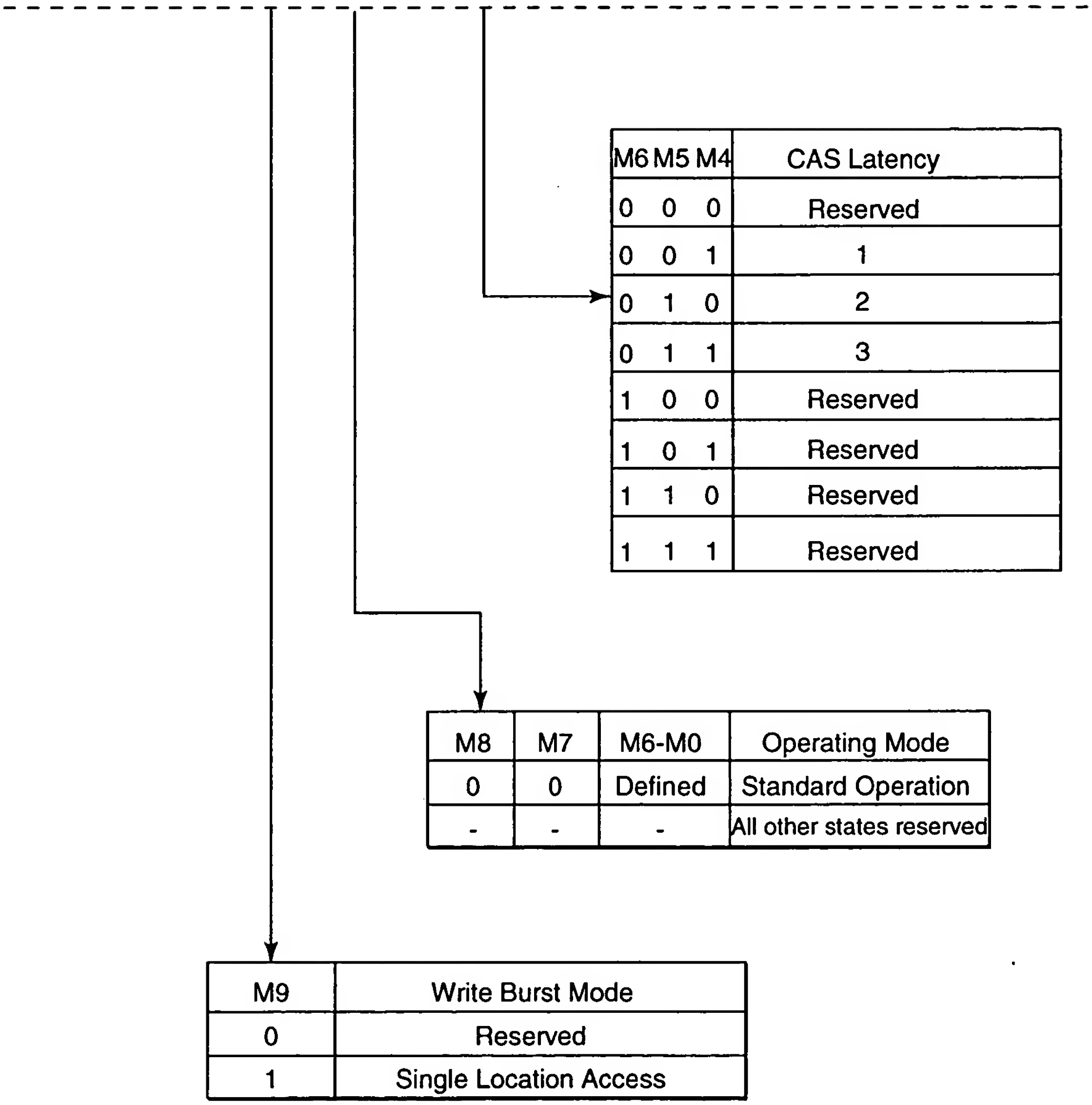
Fig. 2


A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0 Address Bus
11 10 9 8 7 6 5 4 3 2 1 0
Reserved* WB Op Mode CAS Latency BT Burst Length
Mode Register (Mx)
148
*Should program M11, M10 = "0,0" to ensure compatibility with future devices.

M2 M1 M0	Burst Length	
	M3 = 0	M3 = 1
0 0 0	1	1
0 0 1	2	2
0 1 0	4	4
0 1 1	8	8
1 0 0	Reserved	Reserved
1 0 1	Reserved	Reserved
1 1 0	Reserved	Reserved
1 1 1	Full Page	Reserved

M3	Burst Type
0	Sequential
1	Interleaved

Fig. 2A



M6	M5	M4	CAS Latency
0	0	0	Reserved
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

M8	M7	M6-M0	Operating Mode
0	0	Defined	Standard Operation
-	-	-	All other states reserved

M9	Write Burst Mode
0	Reserved
1	Single Location Access

Fig. 2B

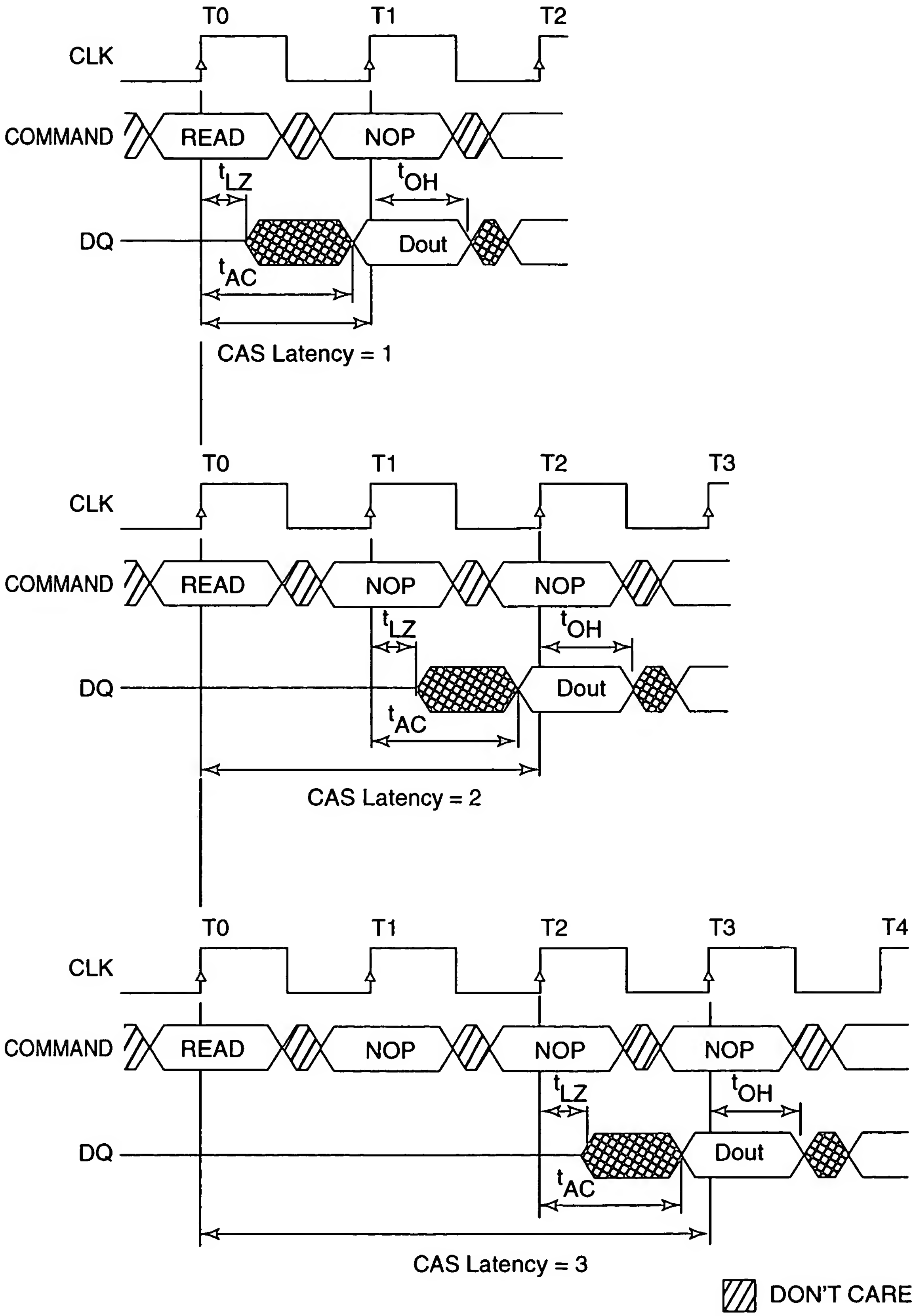

T0
T1
T2
CLK
COMMAND
READ
NOP
t_{LZ}
t_{OH}
DQ
Dout
t_{AC}
CAS Latency = 1
T0
T1
T2
T3
CLK
COMMAND
READ
NOP
NOP
t_{LZ}
t_{OH}
DQ
Dout
t_{AC}
CAS Latency = 2
T0
T1
T2
T3
T4
CLK
COMMAND
READ
NOP
NOP
NOP
t_{LZ}
t_{OH}
DQ
Dout
t_{AC}
CAS Latency = 3
▨ DON'T CARE
▩ UNDEFINED

Fig. 3

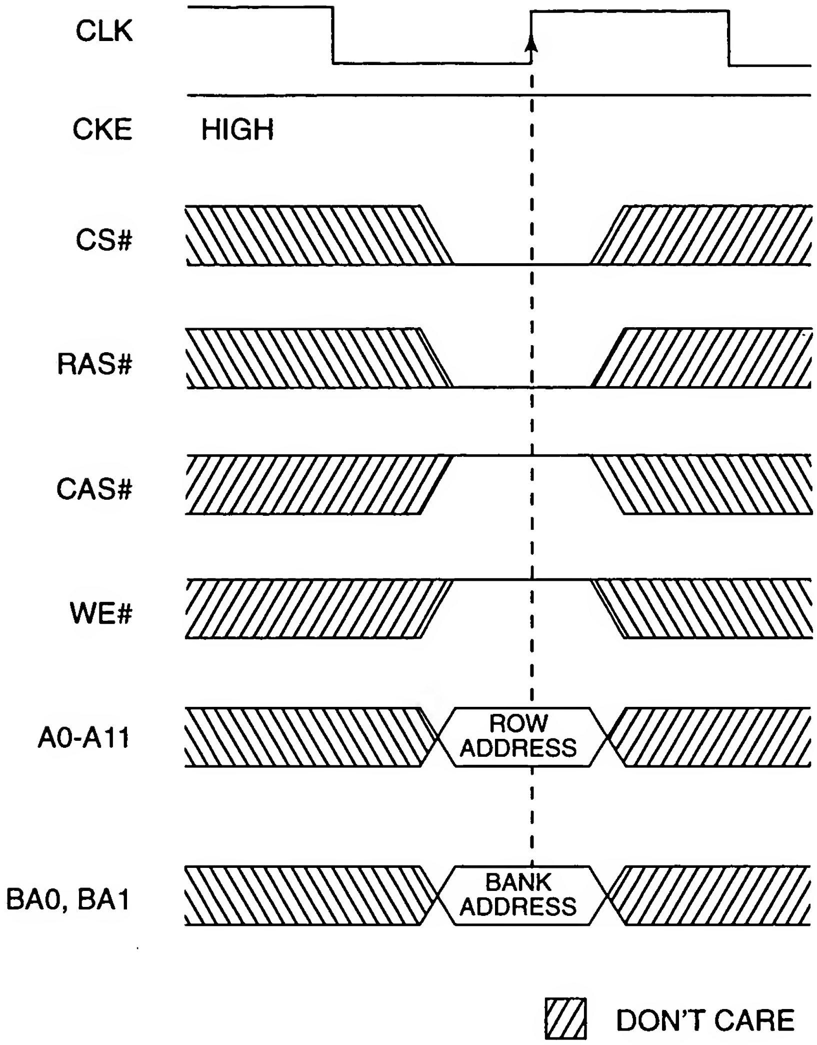

CLK
CKE
HIGH
CS#
RAS#
CAS#
WE#
A0-A11
ROW ADDRESS
BA0, BA1
BANK ADDRESS
DON'T CARE

Fig. 4

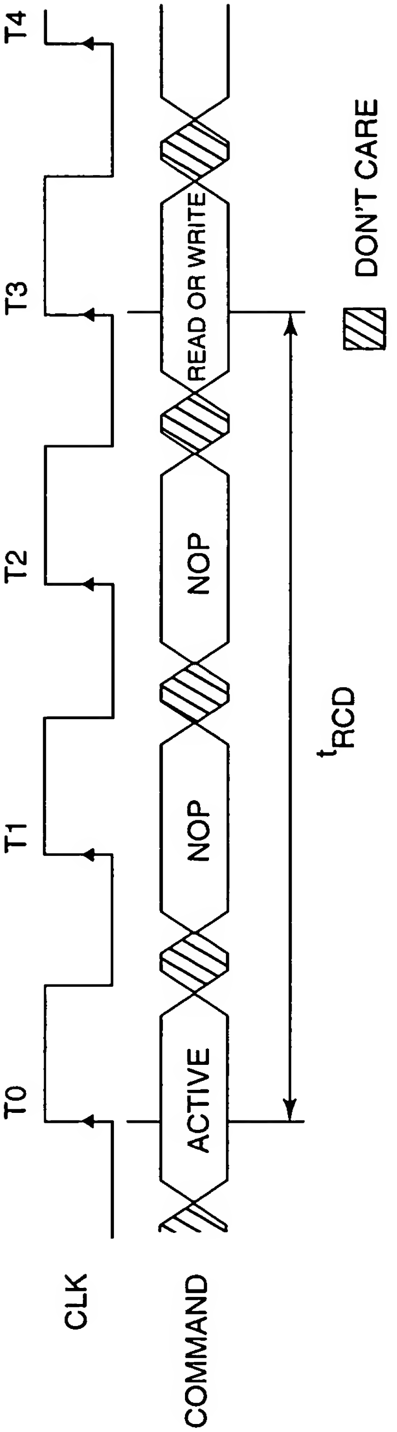

T0
T1
T2
T3
T4
CLK
COMMAND
ACTIVE
NOP
NOP
READ OR WRITE
t_{RCD}
▨ DON'T CARE

Fig. 5

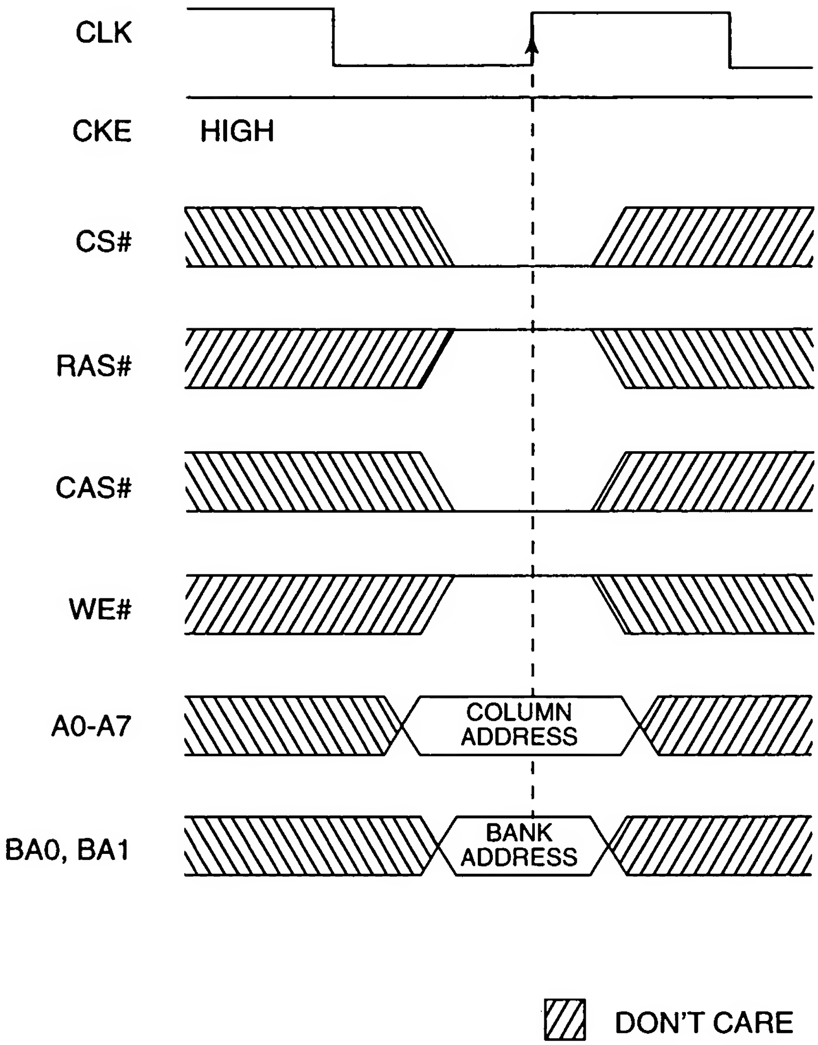
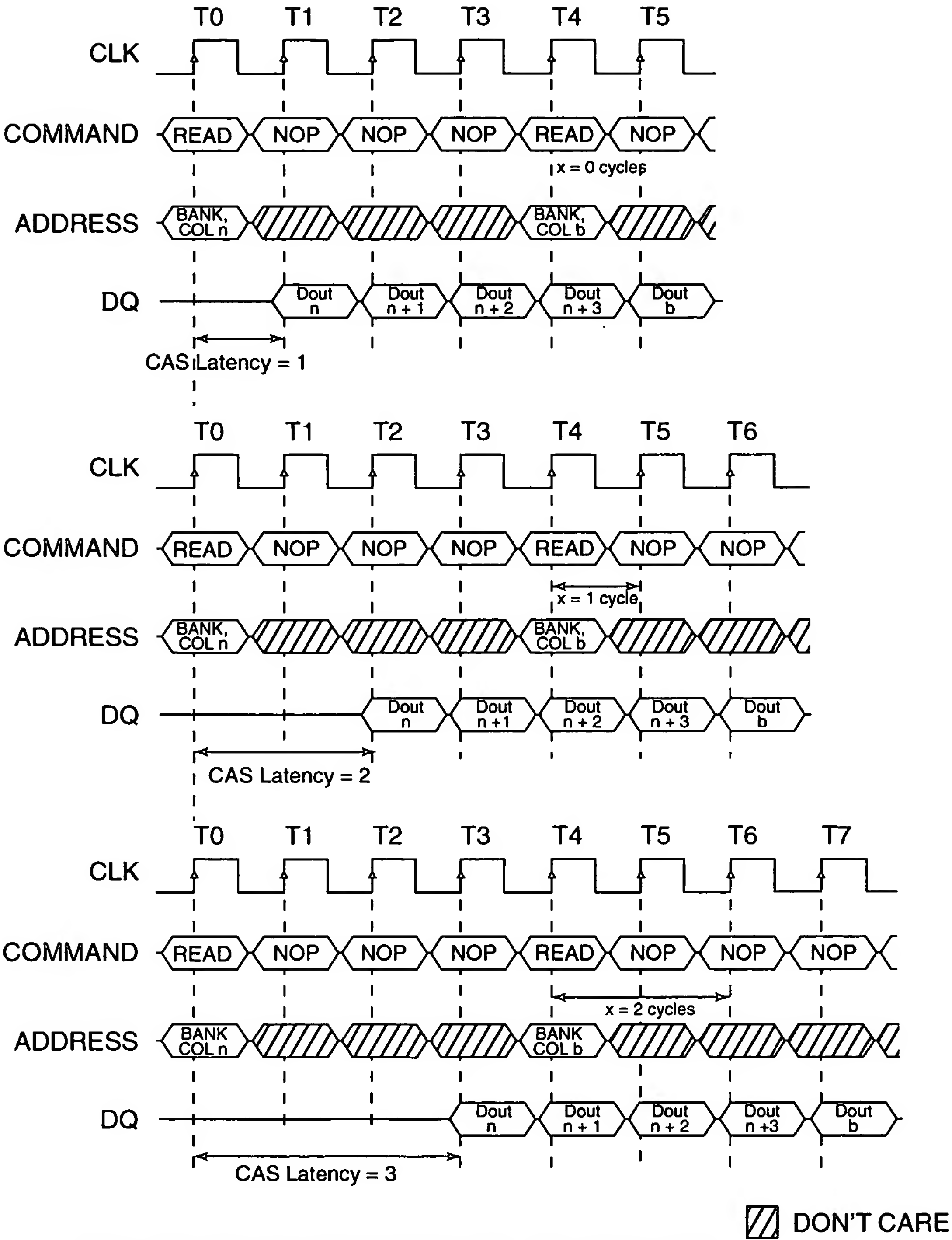
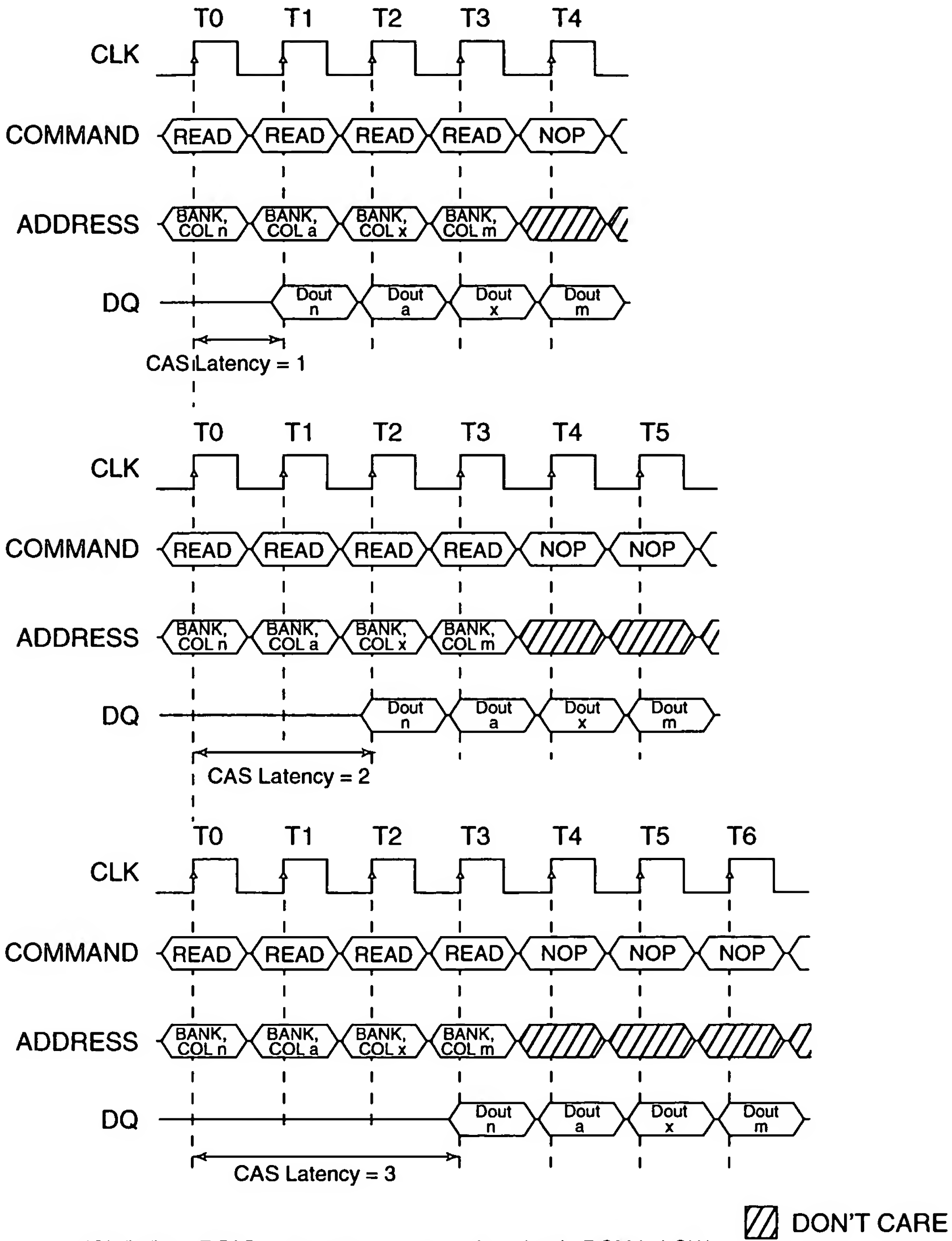

CLK
CKE
HIGH
CS#
RAS#
CAS#
WE#
A0-A7
COLUMN ADDRESS
BA0, BA1
BANK ADDRESS
▨ DON'T CARE

Fig. 6



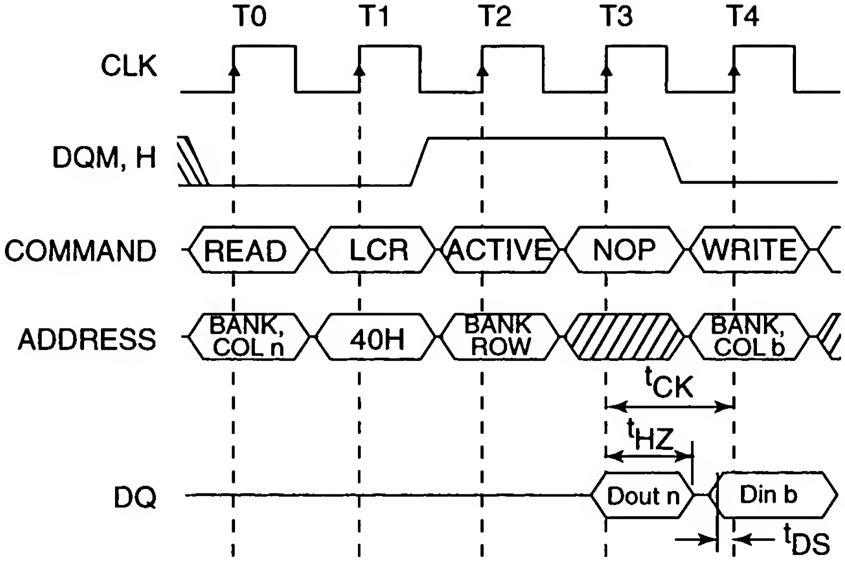
NOTE: Each READ command may be to either bank. DQM is LOW.

Fig. 7



NOTE: Each READ command may be to either bank. DQM is LOW.

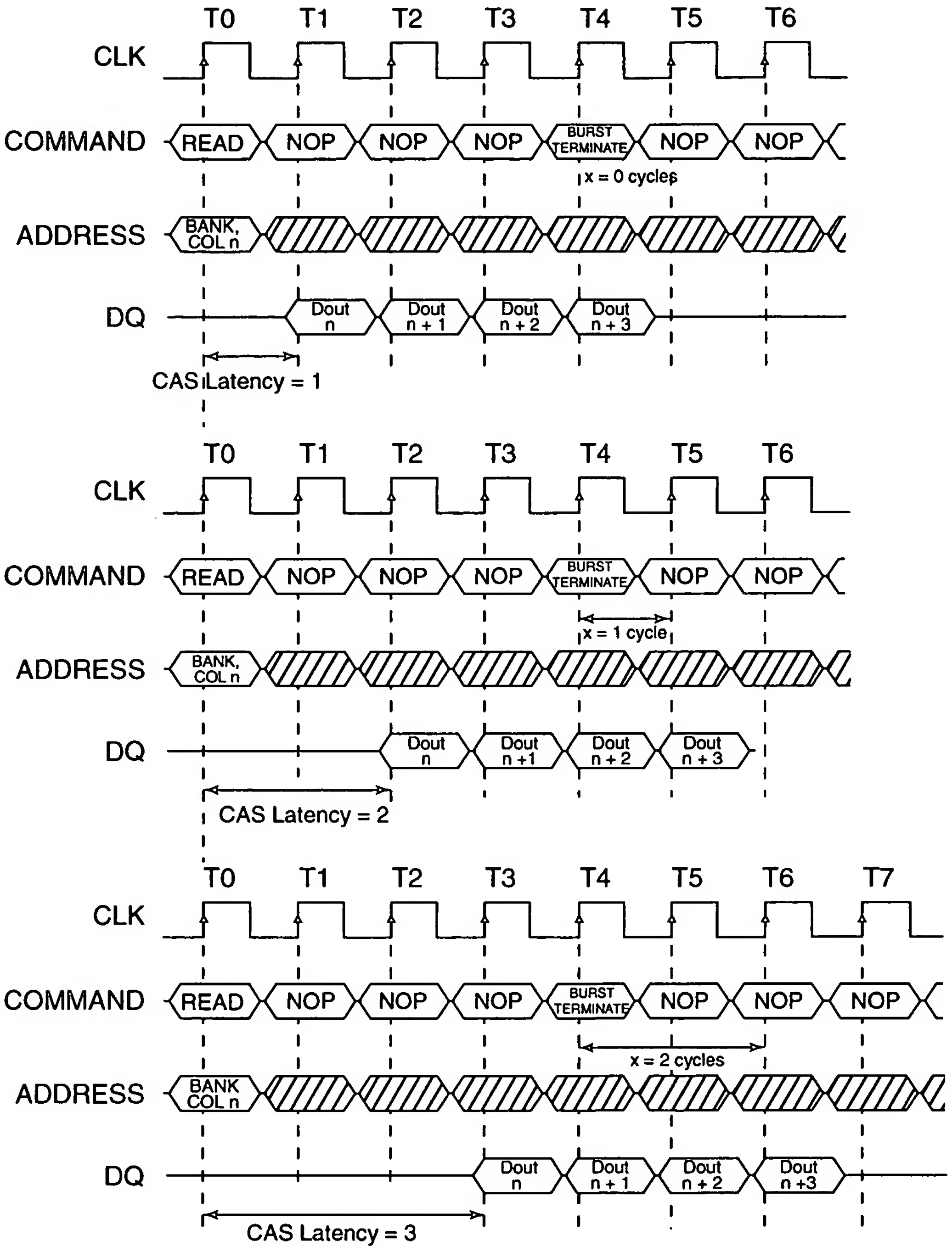
Fig. 8



NOTE: A CAS latency of three is used for illustration. The READ command may be to any bank, and the WRITE command may be to any bank. If a CAS latency of one is used, then DQM is not required.

▨ DON'T CARE

Fig. 9



▨ DON'T CARE

NOTE: DQM is LOW.

Fig. 10

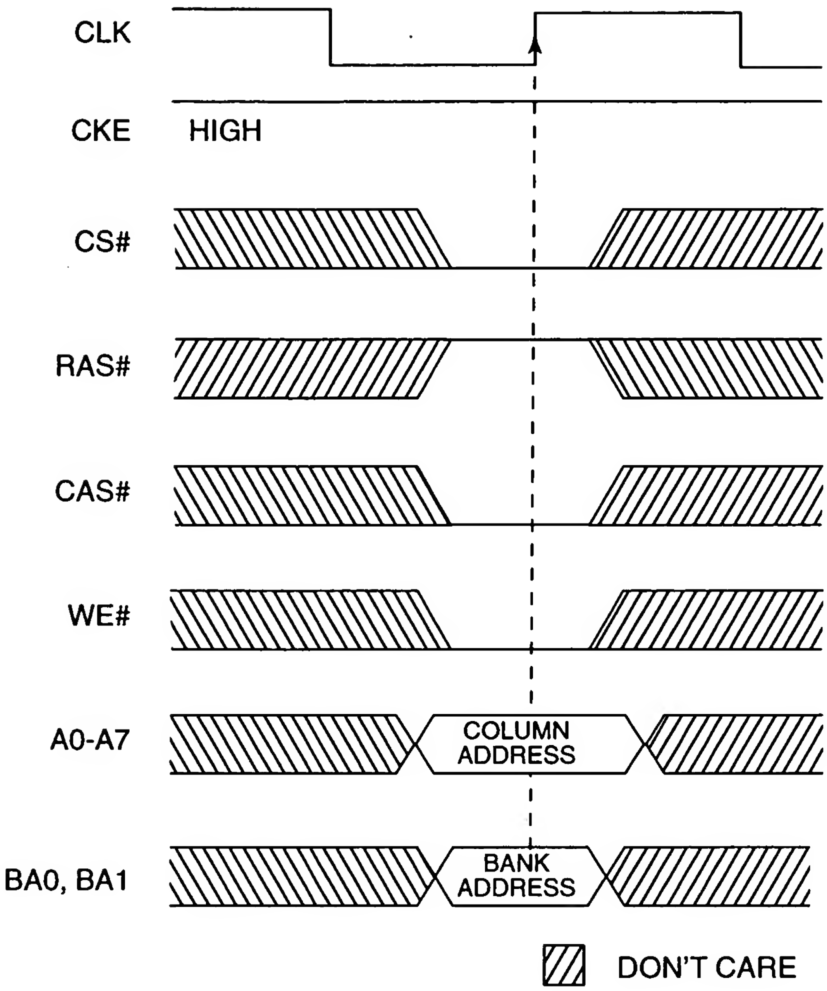
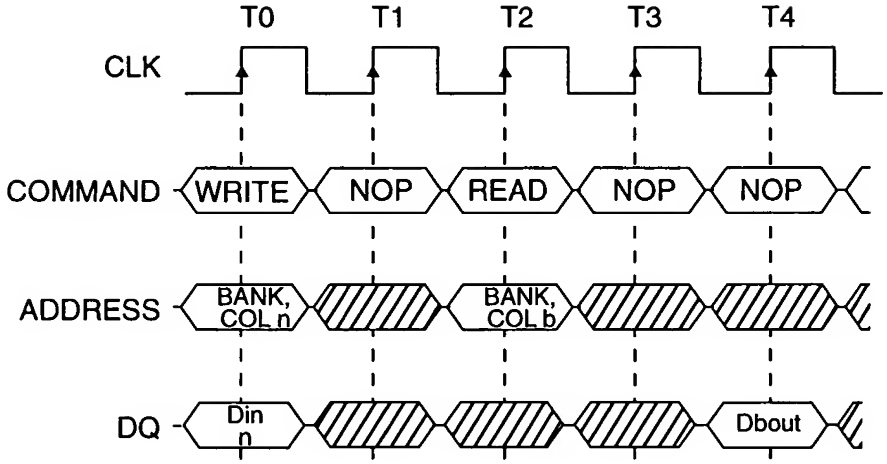

CLK
CKE
HIGH
CS#
RAS#
CAS#
WE#
A0-A7
COLUMN ADDRESS
BA0, BA1
BANK ADDRESS
DON'T CARE

Fig. 11



NOTE: A CAS latency of two is used for illustration. The WRITE command may be to any bank and the READ command may be to any bank. DQM is LOW. A READ to the bank undergoing the WRITE ISM operation may output invalid data.

▨ DON'T CARE

Fig. 12

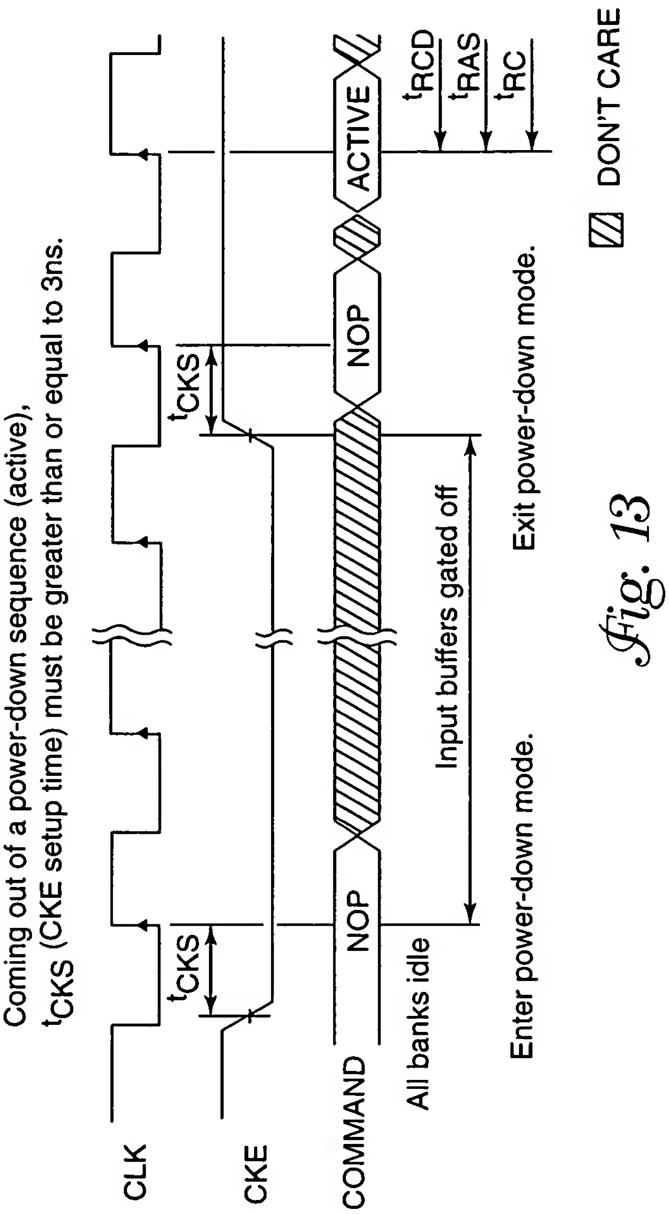
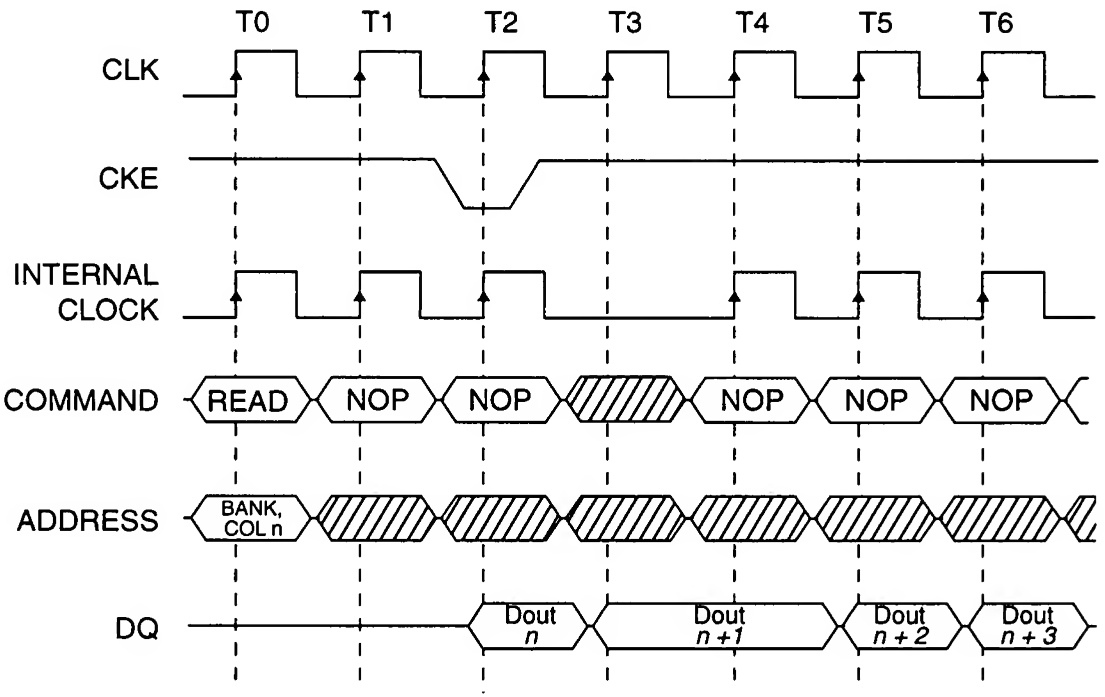

Coming out of a power-down sequence (active),
t_{CKS} (CKE setup time) must be greater than or equal to 3ns.
CLK
CKE
COMMAND
t_{CKS}
t_{CKS}
NOP
NOP
ACTIVE
All banks idle
Input buffers gated off
t_{RCD}
t_{RAS}
t_{RC}
Enter power-down mode.
Exit power-down mode.
▨ DON'T CARE

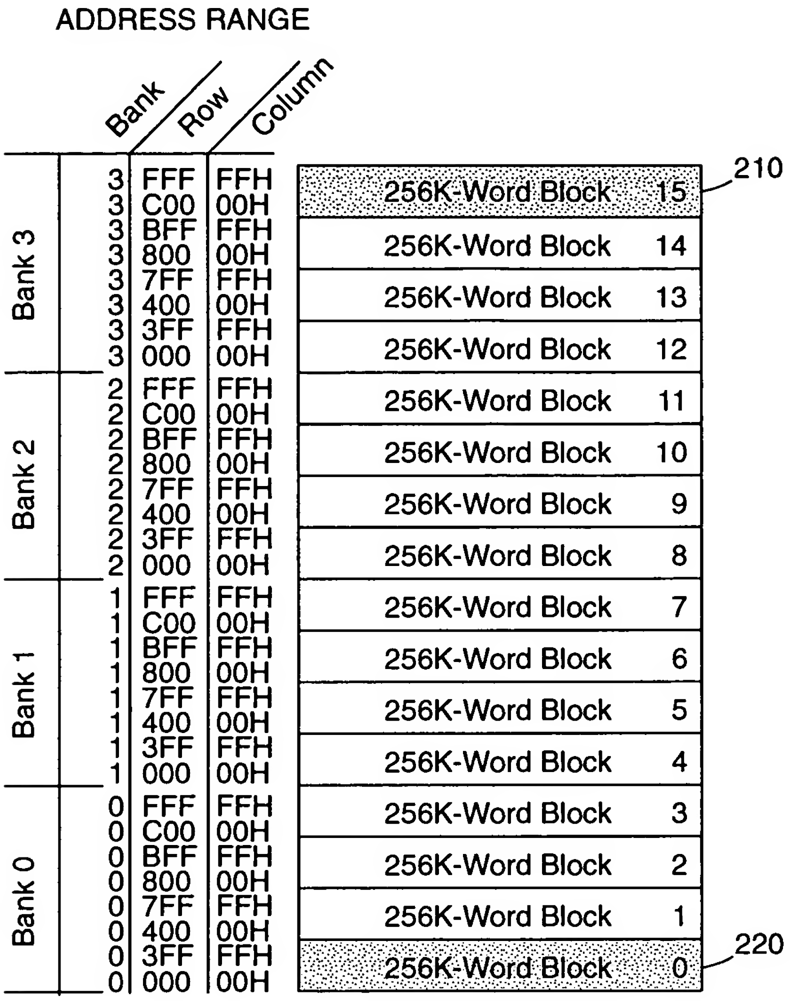
Fig. 13



NOTE: For this example, CAS latency = 2, burst length = 4 or greater, and DQM is LOW.

▨ DON'T CARE

Fig. 14



ADDRESS RANGE

	Bank	Row	Column	
Bank 3	3	FFF	FFH	256K-Word Block 15
	3	C00	00H	
	3	BFF	FFH	256K-Word Block 14
	3	800	00H	
	3	7FF	FFH	256K-Word Block 13
	3	400	00H	
	3	3FF	FFH	256K-Word Block 12
	3	000	00H	
Bank 2	2	FFF	FFH	256K-Word Block 11
	2	C00	00H	
	2	BFF	FFH	256K-Word Block 10
	2	800	00H	
	2	7FF	FFH	256K-Word Block 9
	2	400	00H	
	2	3FF	FFH	256K-Word Block 8
	2	000	00H	
Bank 1	1	FFF	FFH	256K-Word Block 7
	1	C00	00H	
	1	BFF	FFH	256K-Word Block 6
	1	800	00H	
	1	7FF	FFH	256K-Word Block 5
	1	400	00H	
	1	3FF	FFH	256K-Word Block 4
	1	000	00H	
Bank 0	0	FFF	FFH	256K-Word Block 3
	0	C00	00H	
	0	BFF	FFH	256K-Word Block 2
	0	800	00H	
	0	7FF	FFH	256K-Word Block 1
	0	400	00H	
	0	3FF	FFH	256K-Word Block 0
	0	000	00H	

Word-wide (x16)

▨ Software Lock = Hardware-Lock Sectors
RP# = V_{HH} to unprotect if either the
block protect or device protect bit is set.

□ Software Lock = Hardware-Lock Sectors
RP# = V_{CC} to unprotect but must be V_{HH}
if the device protect bit is set.

See BLOCK PROTECT/UNPROTECT SEQUENCE for detailed information.

Fig. 15

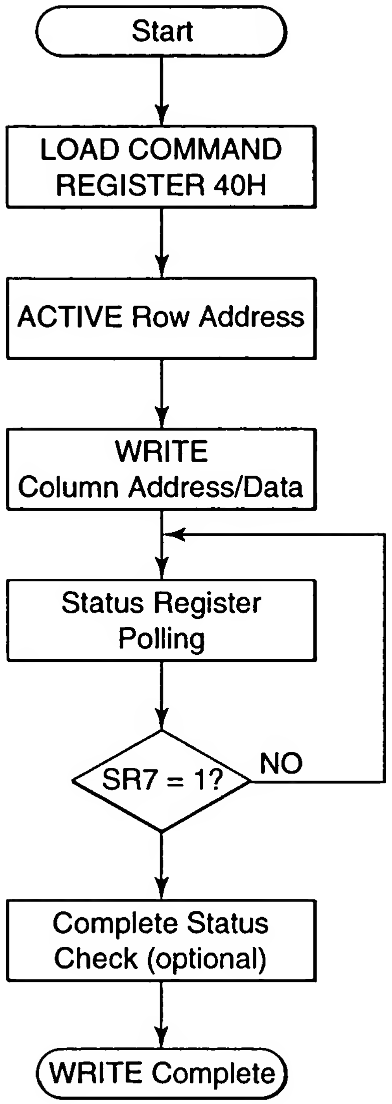

Start
LOAD COMMAND REGISTER 40H
ACTIVE Row Address
WRITE Column Address/Data
Status Register Polling
SR7 = 1?
NO
Complete Status Check (optional)
WRITE Complete

Fig. 16

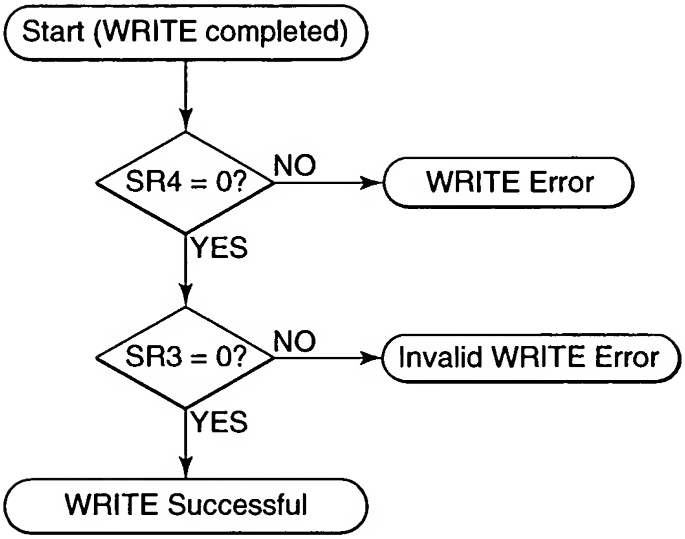

Start (WRITE completed)
SR4 = 0?
NO
WRITE Error
YES
SR3 = 0?
NO
Invalid WRITE Error
YES
WRITE Successful

Fig. 17

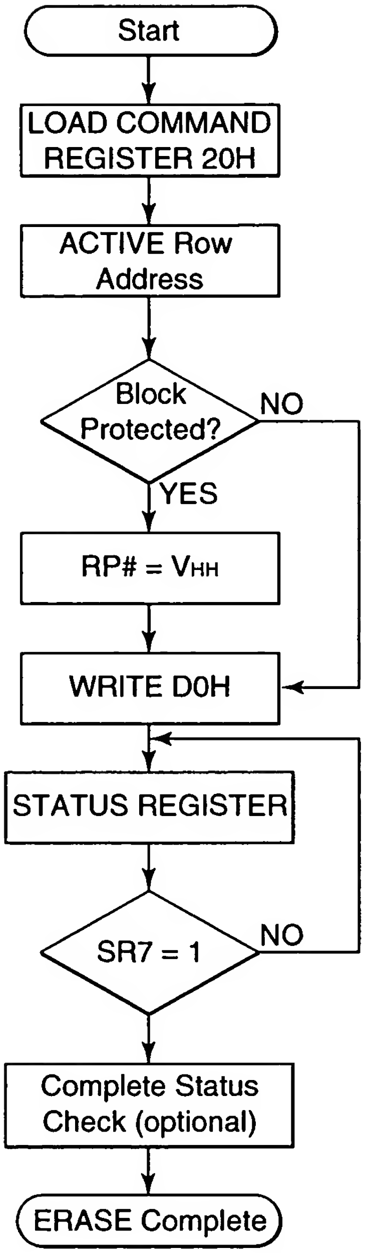

Start
LOAD COMMAND REGISTER 20H
ACTIVE Row Address
Block Protected?
NO
YES
RP# = VHH
WRITE D0H
STATUS REGISTER
SR7 = 1
NO
Complete Status Check (optional)
ERASE Complete

Fig. 18

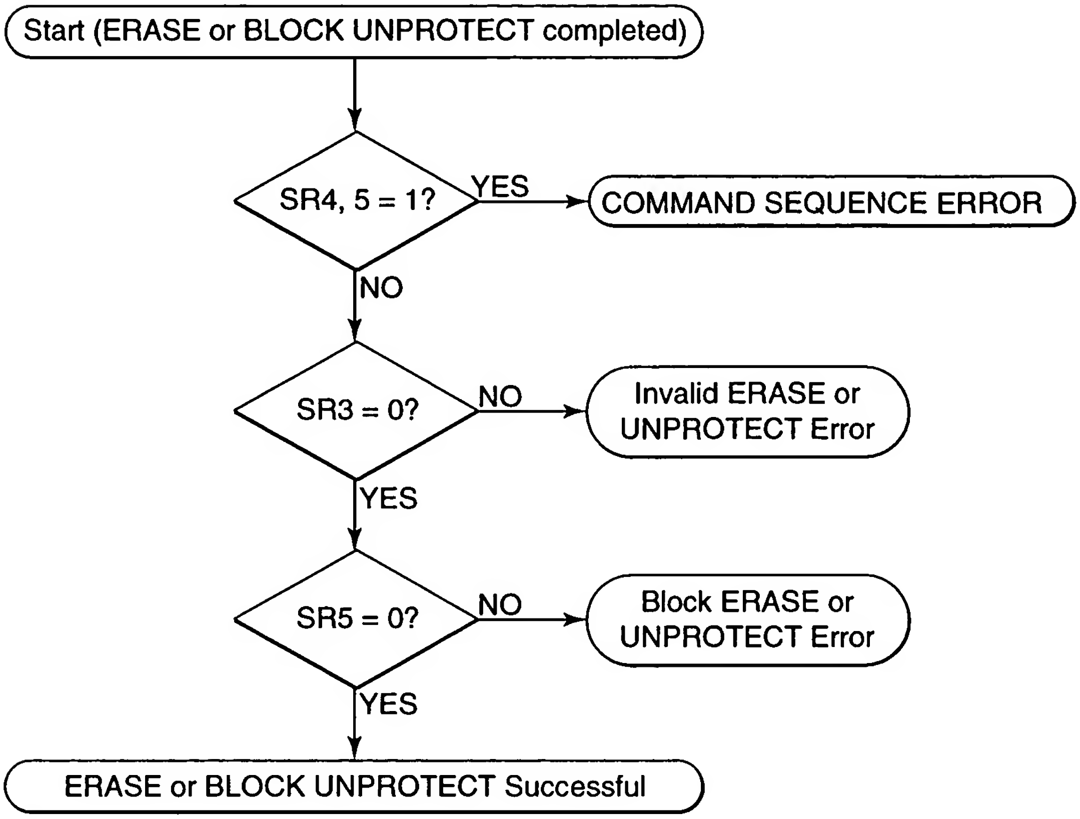

Start (ERASE or BLOCK UNPROTECT completed)
SR4, 5 = 1?
YES
COMMAND SEQUENCE ERROR
NO
SR3 = 0?
NO
Invalid ERASE or UNPROTECT Error
YES
SR5 = 0?
NO
Block ERASE or UNPROTECT Error
YES
ERASE or BLOCK UNPROTECT Successful

Fig. 19

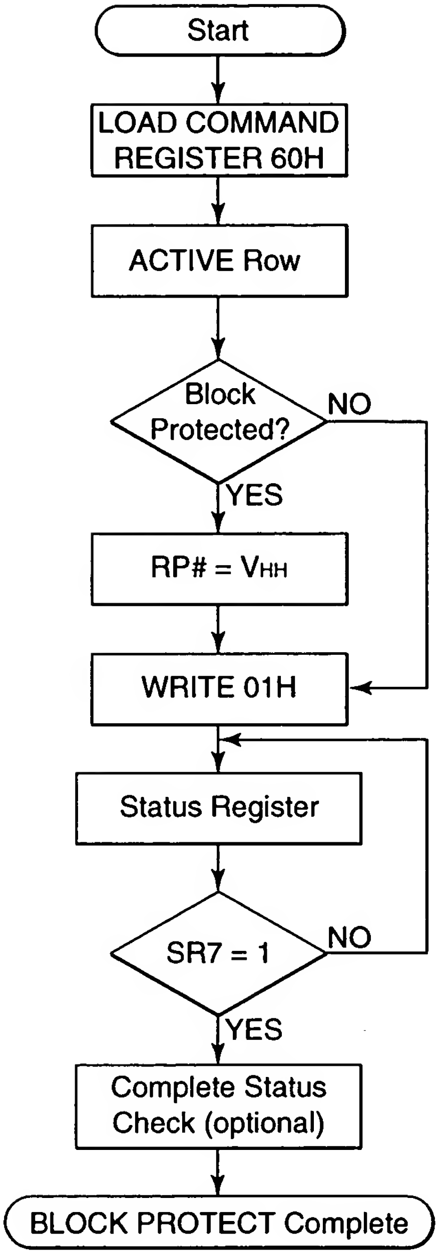

Start
LOAD COMMAND REGISTER 60H
ACTIVE Row
Block Protected?
NO
YES
RP# = VHH
WRITE 01H
Status Register
SR7 = 1
NO
YES
Complete Status Check (optional)
BLOCK PROTECT Complete

Fig. 20

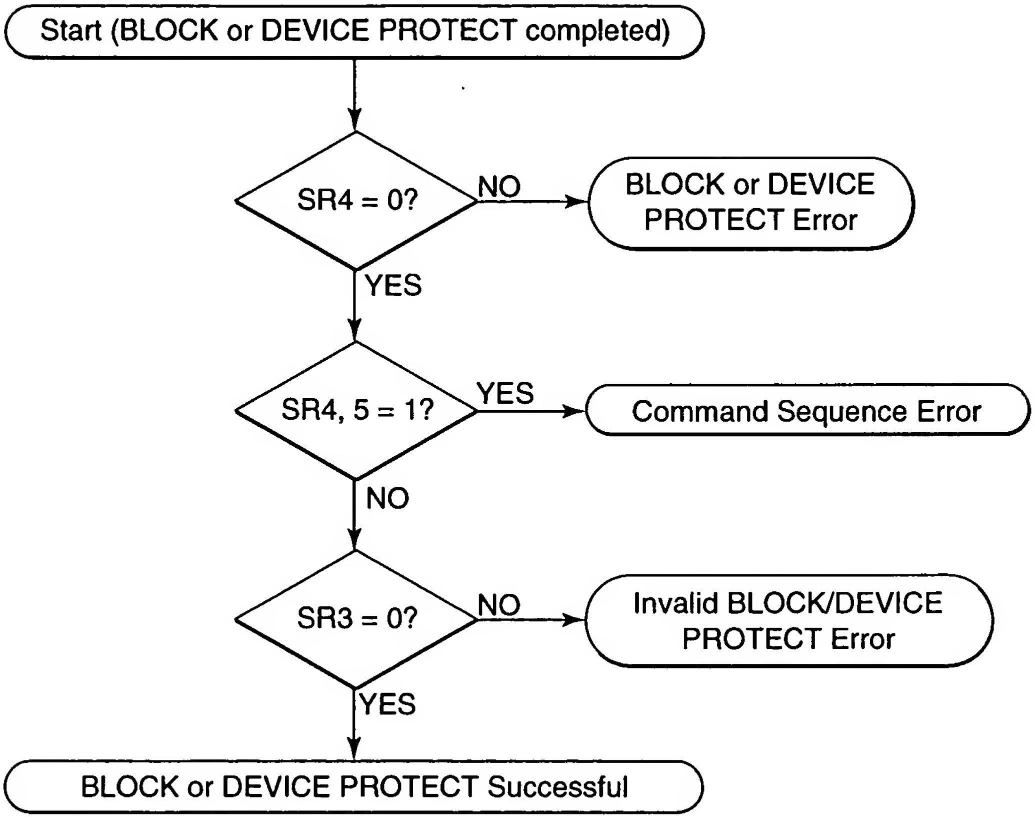

Start (BLOCK or DEVICE PROTECT completed)
SR4 = 0?
NO
BLOCK or DEVICE PROTECT Error
YES
SR4, 5 = 1?
YES
Command Sequence Error
NO
SR3 = 0?
NO
Invalid BLOCK/DEVICE PROTECT Error
YES
BLOCK or DEVICE PROTECT Successful

Fig. 21

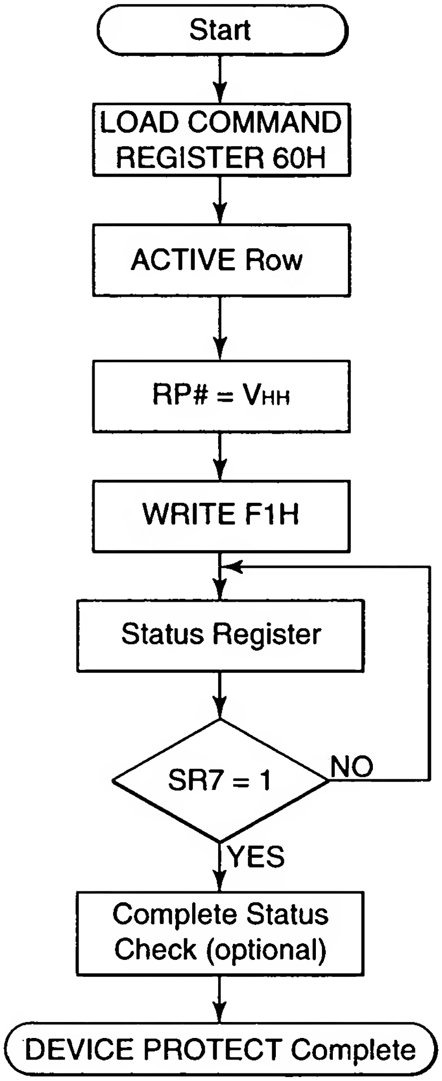

Start
LOAD COMMAND REGISTER 60H
ACTIVE Row
RP# = VHH
WRITE F1H
Status Register
SR7 = 1
NO
YES
Complete Status Check (optional)
DEVICE PROTECT Complete

Fig. 22

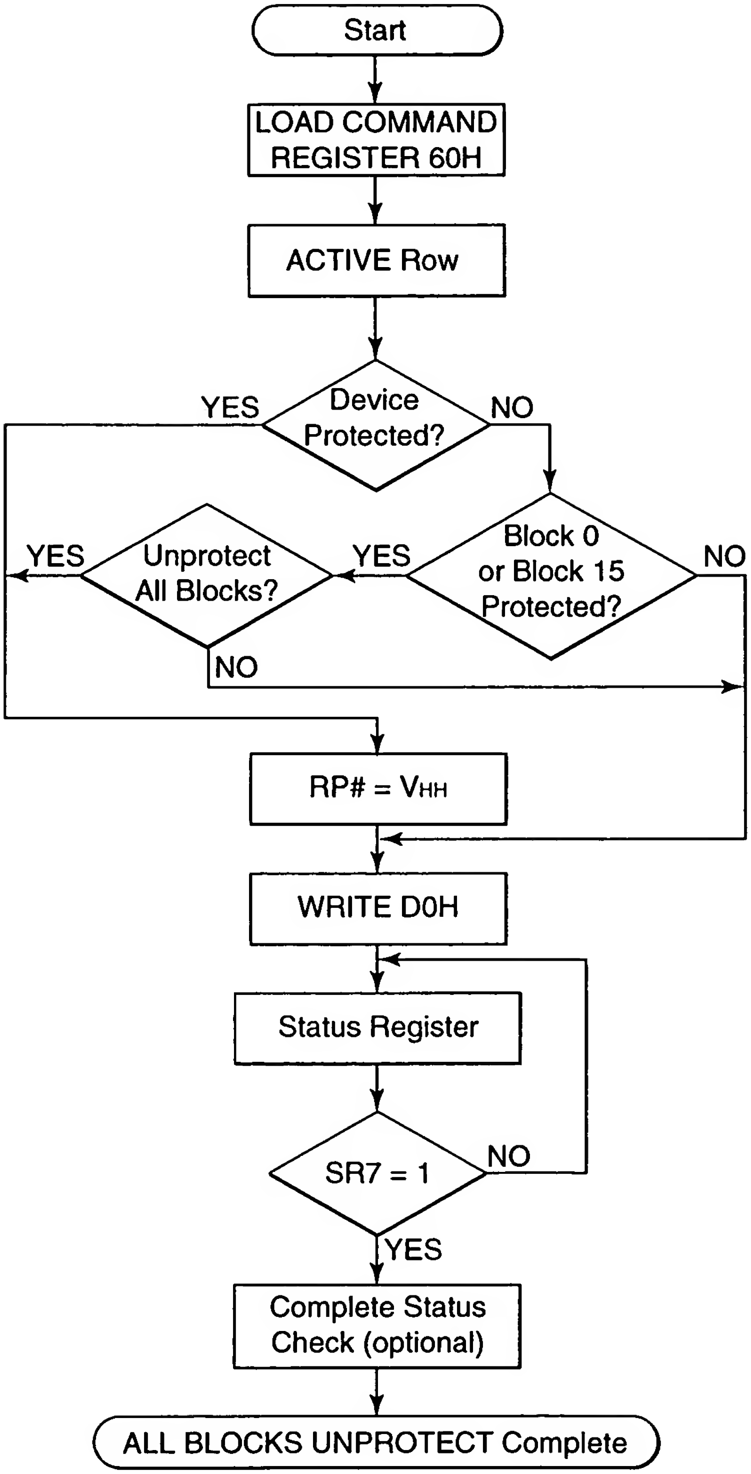

Start
LOAD COMMAND REGISTER 60H
ACTIVE Row
Device Protected?
YES
NO
Unprotect All Blocks?
YES
YES
Block 0 or Block 15 Protected?
NO
NO
RP# = VHH
WRITE D0H
Status Register
SR7 = 1
NO
YES
Complete Status Check (optional)
ALL BLOCKS UNPROTECT Complete

Fig. 23

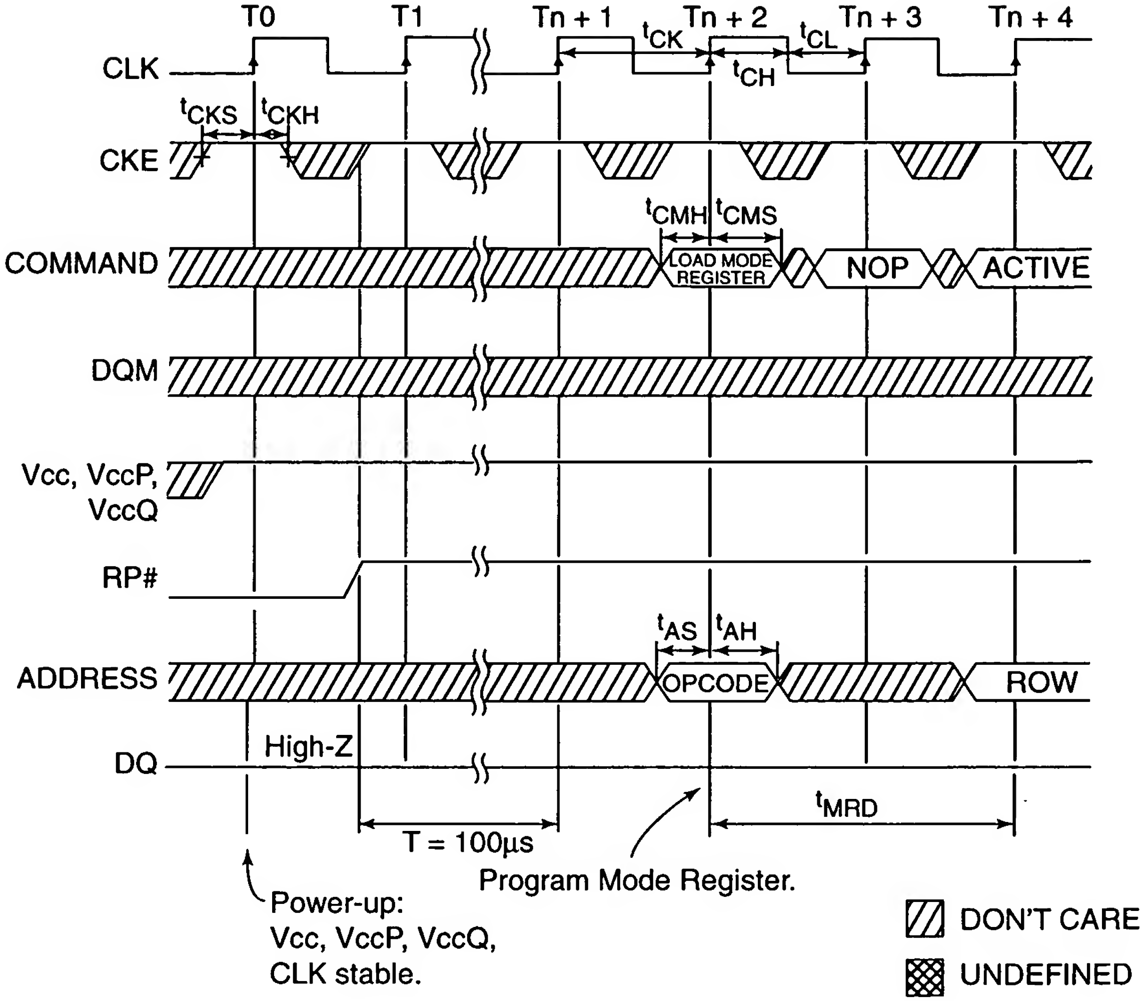

T0
T1
Tn + 1
Tn + 2
Tn + 3
Tn + 4
t_{CK}
t_{CL}
t_{CH}
CLK
t_{CKS}
t_{CKH}
CKE
t_{CMH}
t_{CMS}
COMMAND
LOAD MODE REGISTER
NOP
ACTIVE
DQM
Vcc, VccP, VccQ
RP#
t_{AS}
t_{AH}
ADDRESS
OPCODE
ROW
DQ
High-Z
t_{MRD}
T = 100µs
Program Mode Register.
Power-up: Vcc, VccP, VccQ, CLK stable.
▨ DON'T CARE
▩ UNDEFINED

Fig. 24

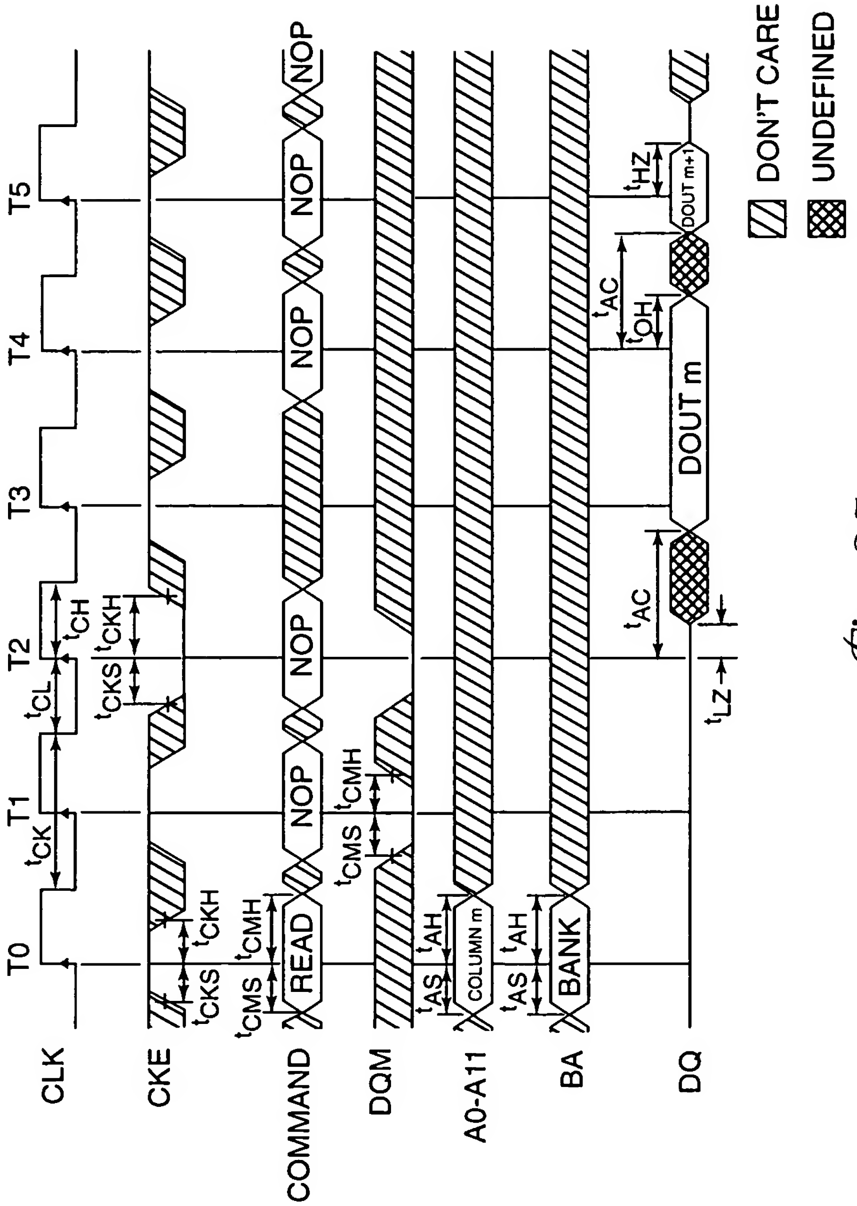

CLK
CKE
COMMAND
DQM
A0-A11
BA
DQ
T0
T1
T2
T3
T4
T5
t_{CK}
t_{CL}
t_{CH}
t_{CKS}
t_{CKH}
t_{CMS}
t_{CMH}
t_{AS}
t_{AH}
t_{AC}
t_{LZ}
t_{OH}
t_{HZ}
READ
NOP
NOP
NOP
NOP
NOP
NOP
COLUMN m
BANK
DOUT m
DOUT m+1
DON'T CARE
UNDEFINED

Fig. 25

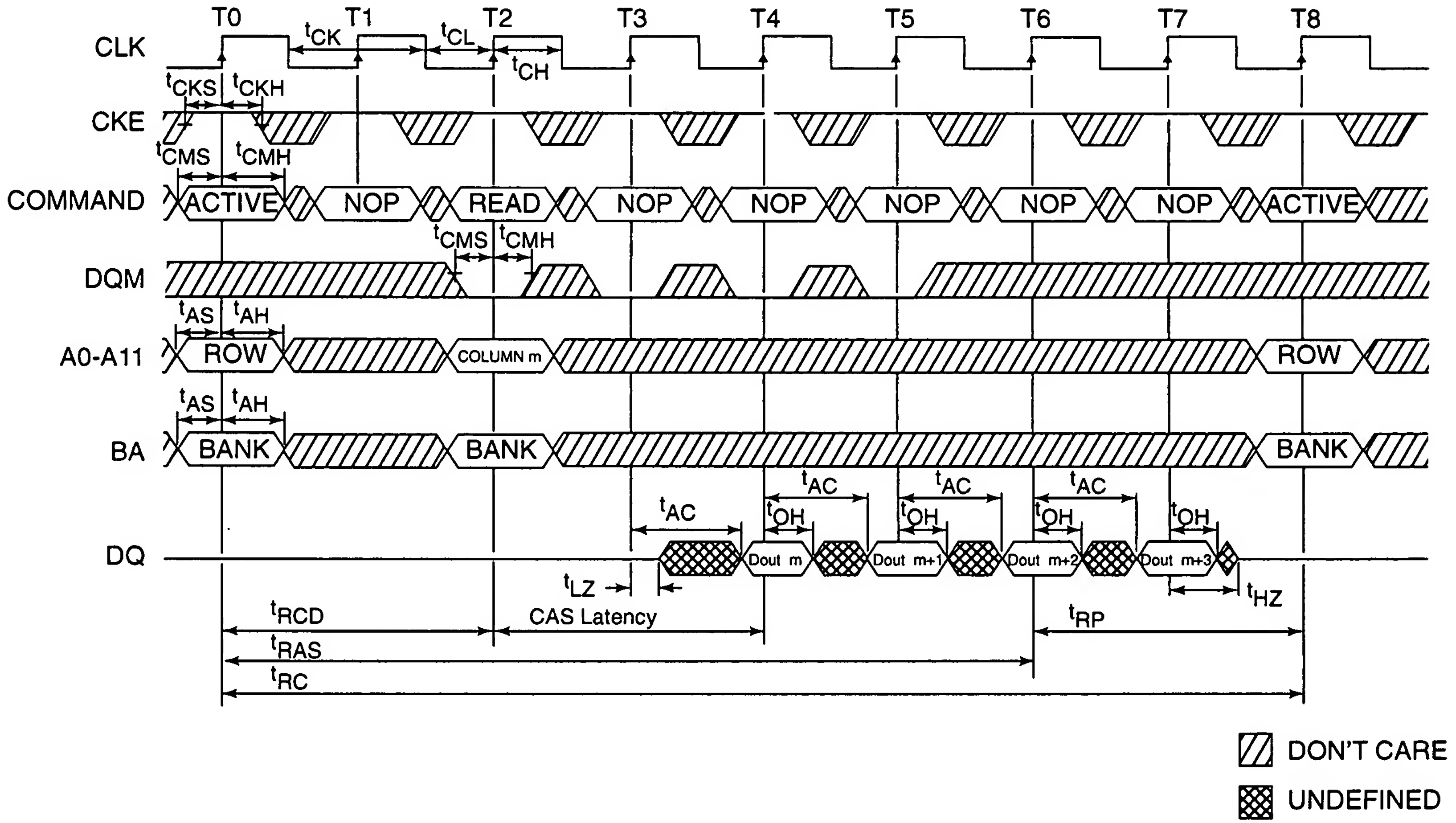

CLK
CKE
COMMAND
DQM
A0-A11
BA
DQ
T0
T1
T2
T3
T4
T5
T6
T7
T8
t_{CK}
t_{CL}
t_{CH}
t_{CKS}
t_{CKH}
t_{CMS}
t_{CMH}
ACTIVE
NOP
READ
NOP
NOP
NOP
NOP
NOP
ACTIVE
t_{CMS}
t_{CMH}
t_{AS}
t_{AH}
ROW
COLUMN m
ROW
t_{AS}
t_{AH}
BANK
BANK
BANK
t_{AC}
t_{AC}
t_{AC}
t_{AC}
t_{OH}
t_{OH}
t_{OH}
t_{OH}
Dout m
Dout m+1
Dout m+2
Dout m+3
t_{LZ}
t_{HZ}
t_{RCD}
CAS Latency
t_{RP}
t_{RAS}
t_{RC}
▨ DON'T CARE
▩ UNDEFINED

Fig. 26

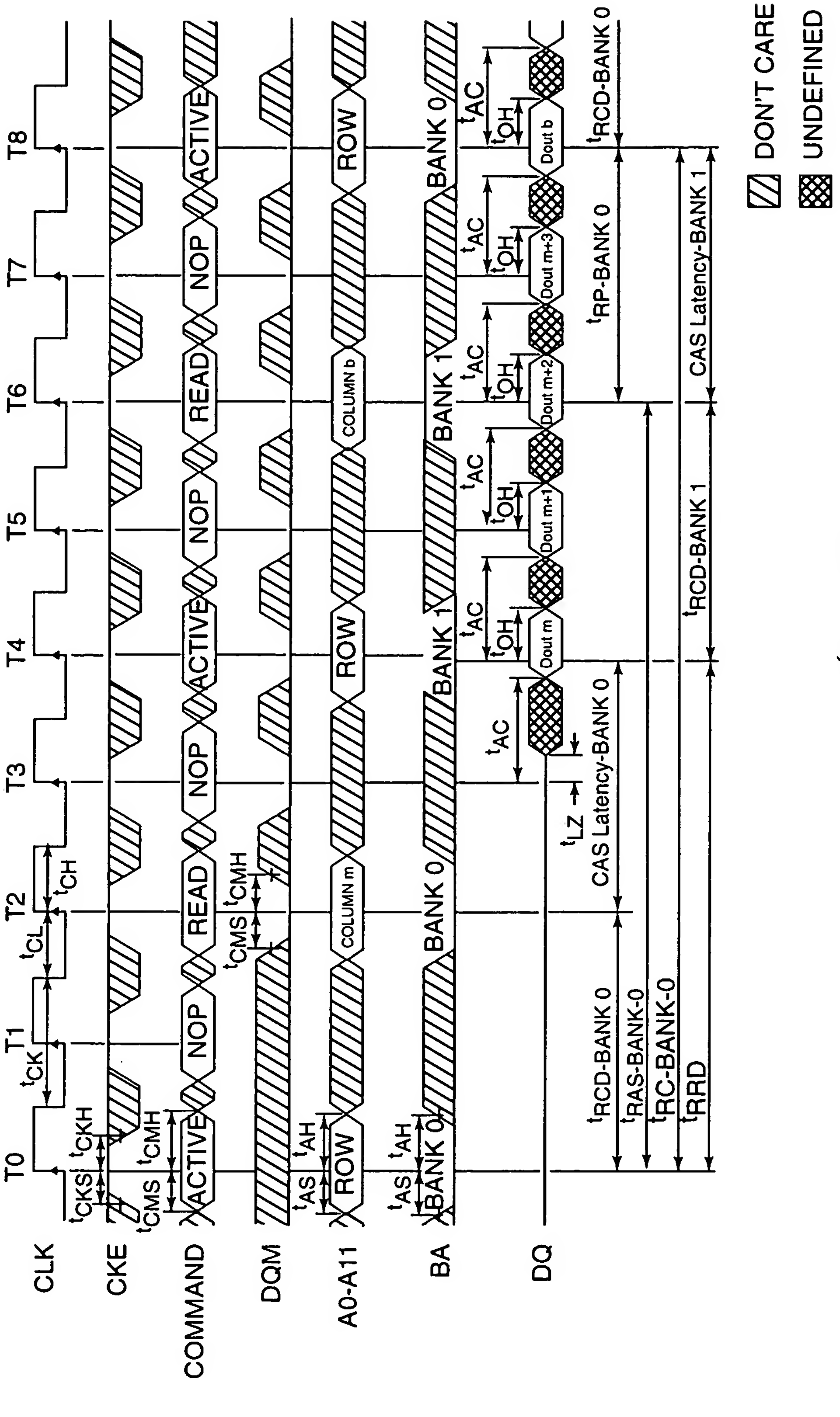

CLK
CKE
COMMAND
DQM
A0-A11
BA
DQ
T0
T1
T2
T3
T4
T5
T6
T7
T8
tCKS
tCKH
tCK
tCL
tCH
tCMS
tCMH
tAS
tAH
ACTIVE
NOP
READ
NOP
ACTIVE
NOP
READ
NOP
ACTIVE
ROW
COLUMN m
ROW
COLUMN b
ROW
BANK 0
BANK 0
BANK 1
BANK 1
BANK 0
tAC
tOH
tLZ
Dout m
Dout m+1
Dout m+2
Dout m+3
Dout b
tRCD-BANK 0
tRAS-BANK-0
tRC-BANK-0
tRRD
CAS Latency-BANK 0
tRCD-BANK 1
tRP-BANK 0
CAS Latency-BANK 1
▨ DON'T CARE
▩ UNDEFINED

Fig. 27

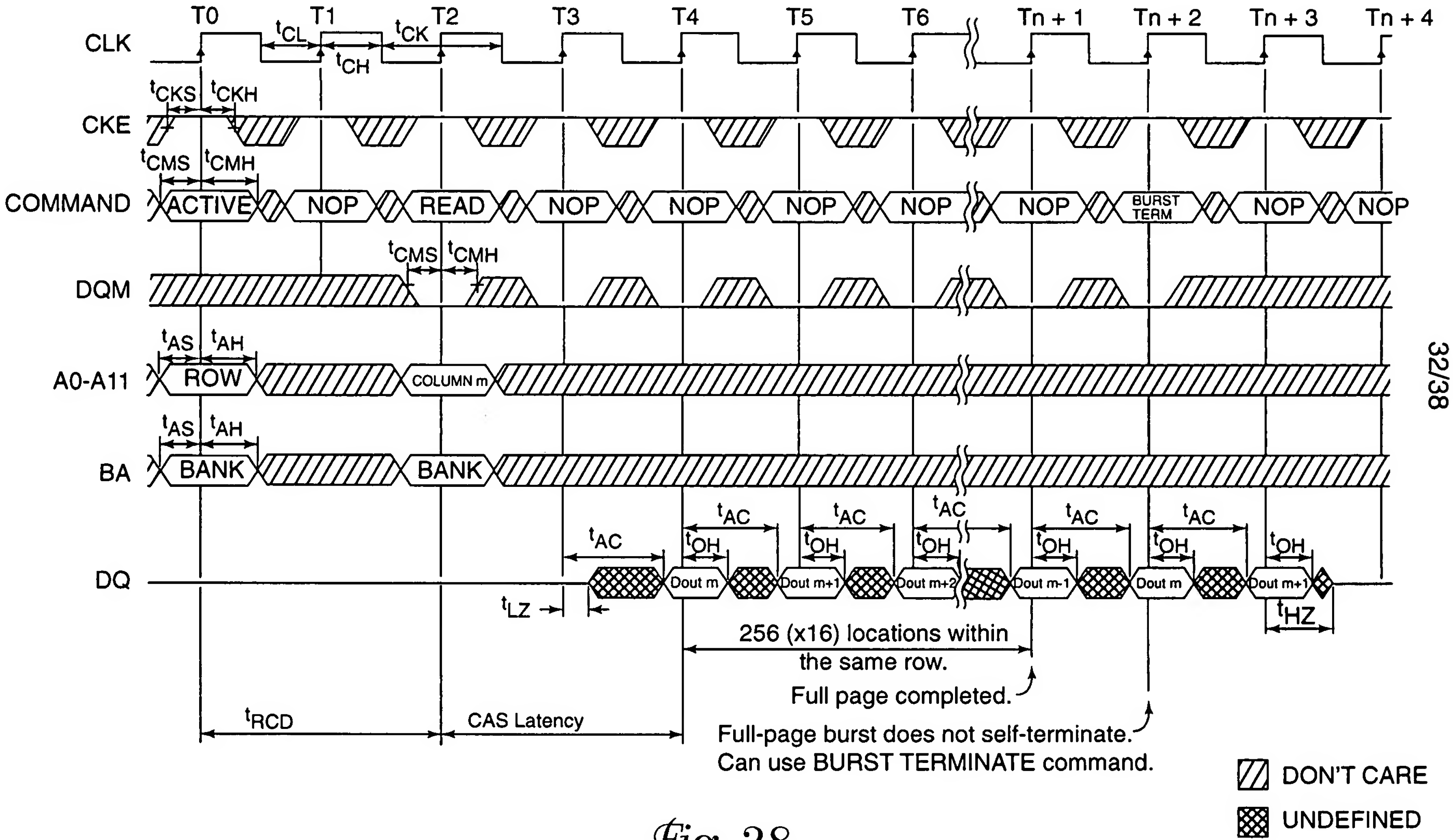

CLK
CKE
COMMAND
DQM
A0-A11
BA
DQ
T0
T1
T2
T3
T4
T5
T6
Tn + 1
Tn + 2
Tn + 3
Tn + 4
t_{CL}
t_{CH}
t_{CK}
t_{CKS}
t_{CKH}
t_{CMS}
t_{CMH}
ACTIVE
NOP
READ
NOP
NOP
NOP
NOP
NOP
BURST TERM
NOP
NOP
t_{CMS}
t_{CMH}
t_{AS}
t_{AH}
ROW
COLUMN m
t_{AS}
t_{AH}
BANK
BANK
t_{AC}
t_{OH}
t_{LZ}
Dout m
Dout m+1
Dout m+2
Dout m-1
Dout m
Dout m+1
t_{HZ}
256 (x16) locations within the same row.
Full page completed.
Full-page burst does not self-terminate. Can use BURST TERMINATE command.
t_{RCD}
CAS Latency
▨ DON'T CARE
▩ UNDEFINED

Fig. 28

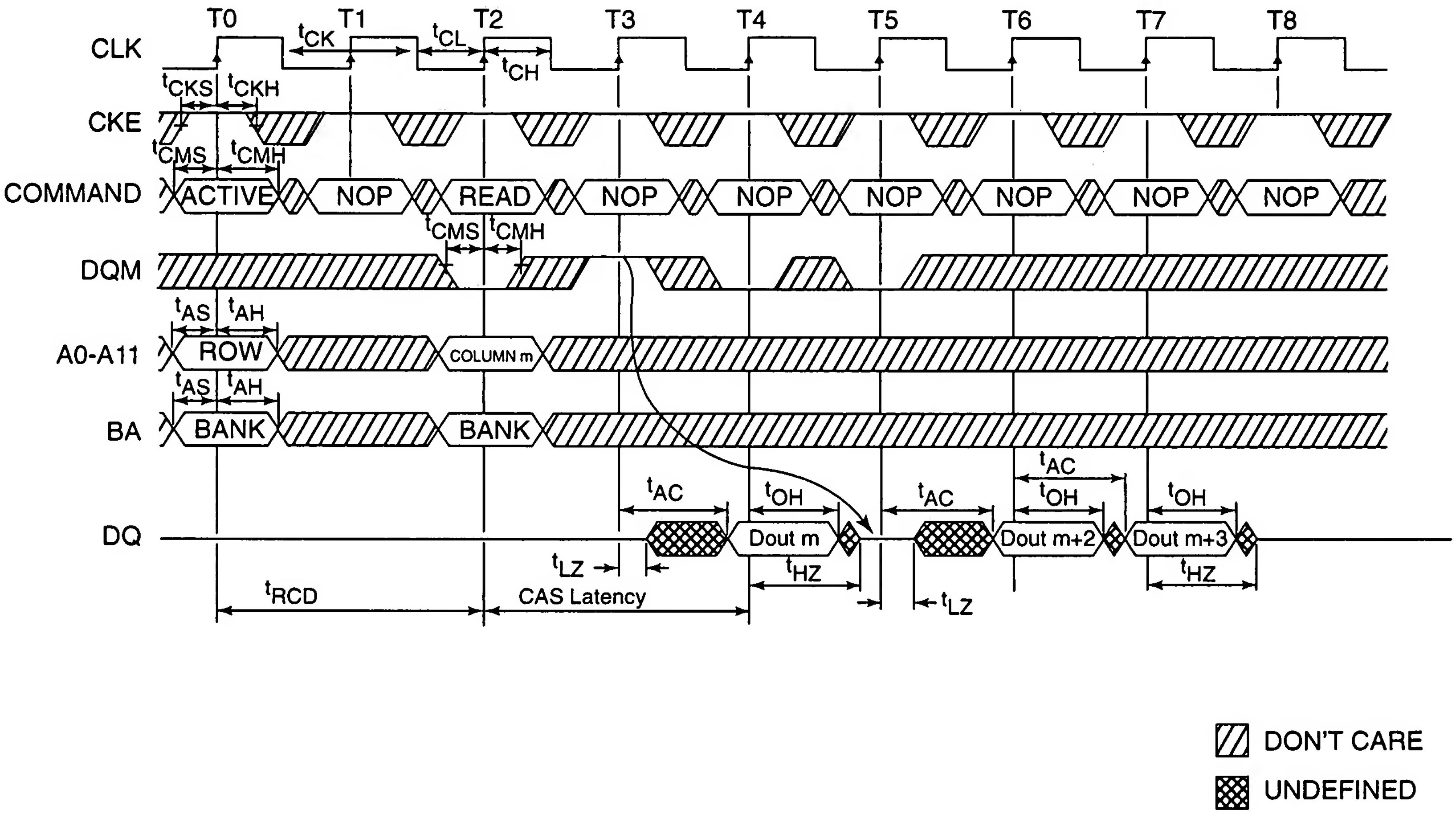

T0
T1
T2
T3
T4
T5
T6
T7
T8
CLK
t_{CK}
t_{CL}
t_{CH}
t_{CKS}
t_{CKH}
CKE
t_{CMS}
t_{CMH}
COMMAND
ACTIVE
NOP
READ
NOP
NOP
NOP
NOP
NOP
NOP
t_{CMS}
t_{CMH}
DQM
t_{AS}
t_{AH}
A0-A11
ROW
COLUMN m
t_{AS}
t_{AH}
BA
BANK
BANK
DQ
t_{AC}
t_{OH}
Dout m
t_{AC}
t_{AC}
t_{OH}
Dout m+2
t_{OH}
Dout m+3
t_{LZ}
t_{HZ}
t_{LZ}
t_{HZ}
t_{RCD}
CAS Latency
▨ DON'T CARE
▩ UNDEFINED

Fig. 29

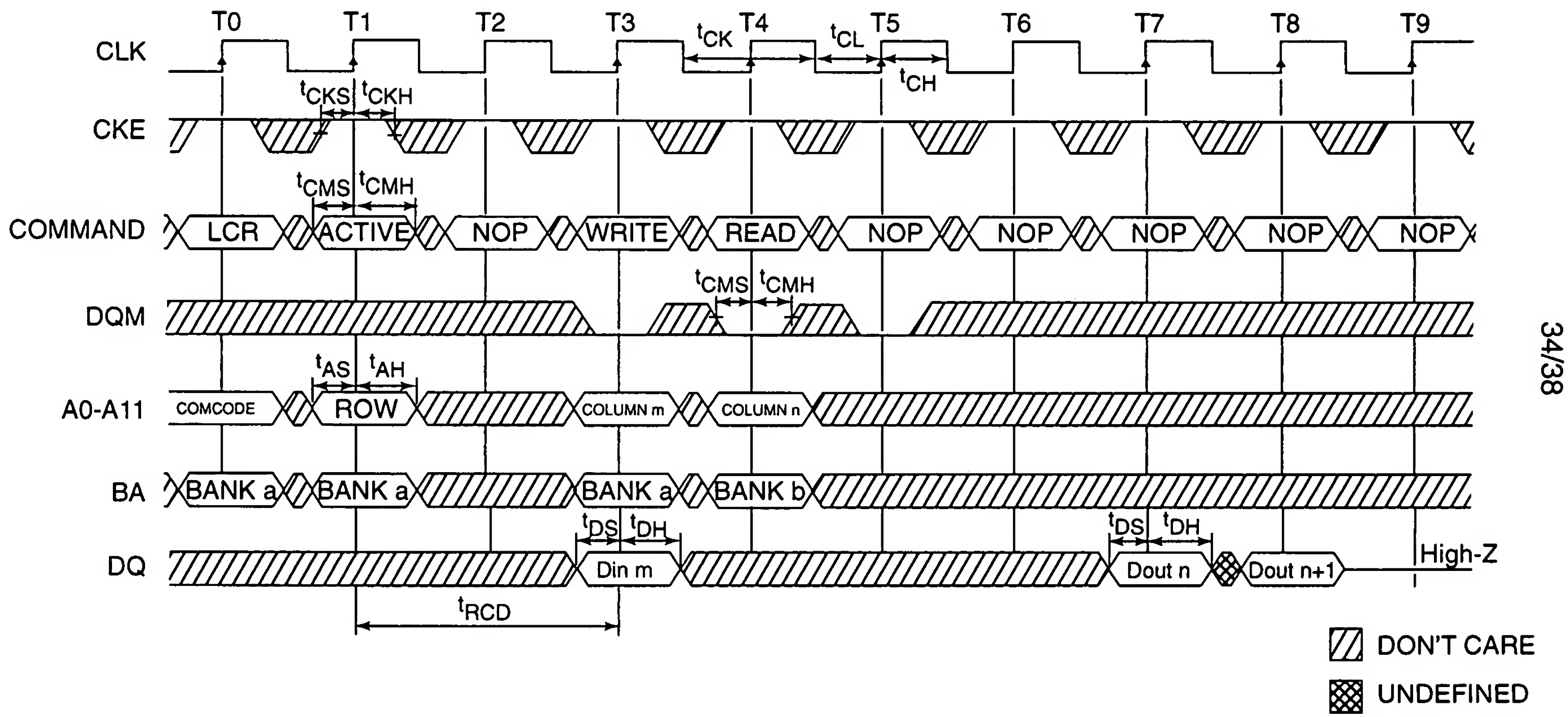

CLK
CKE
COMMAND
DQM
A0-A11
BA
DQ
T0
T1
T2
T3
T4
T5
T6
T7
T8
T9
t_{CK}
t_{CL}
t_{CH}
t_{CKS}
t_{CKH}
t_{CMS}
t_{CMH}
LCR
ACTIVE
NOP
WRITE
READ
NOP
NOP
NOP
NOP
NOP
t_{CMS}
t_{CMH}
t_{AS}
t_{AH}
COMCODE
ROW
COLUMN m
COLUMN n
BANK a
BANK a
BANK a
BANK b
t_{DS}
t_{DH}
Din m
t_{DS}
t_{DH}
Dout n
Dout n+1
High-Z
t_{RCD}
DON'T CARE
UNDEFINED

Fig. 30

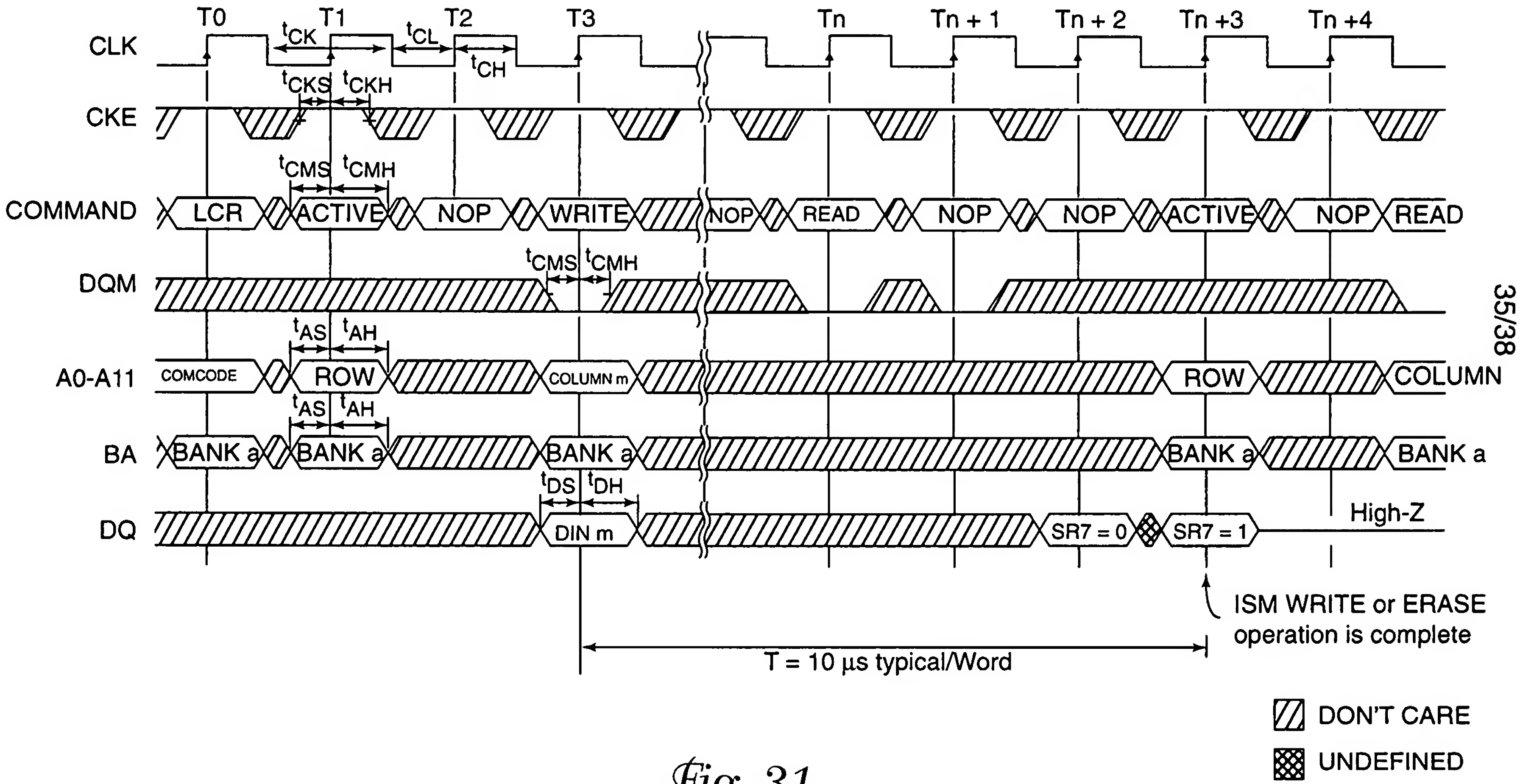

CLK
CKE
COMMAND
DQM
A0-A11
BA
DQ
T0
T1
T2
T3
Tn
Tn + 1
Tn + 2
Tn +3
Tn +4
tCK
tCL
tCH
tCKS
tCKH
tCMS
tCMH
tAS
tAH
tDS
tDH
LCR
ACTIVE
NOP
WRITE
NOP
READ
NOP
NOP
ACTIVE
NOP
READ
COMCODE
ROW
COLUMN m
ROW
COLUMN
BANK a
BANK a
BANK a
BANK a
BANK a
DIN m
SR7 = 0
SR7 = 1
High-Z
ISM WRITE or ERASE operation is complete
T = 10 µs typical/Word
▨ DON'T CARE
▩ UNDEFINED

Fig. 31

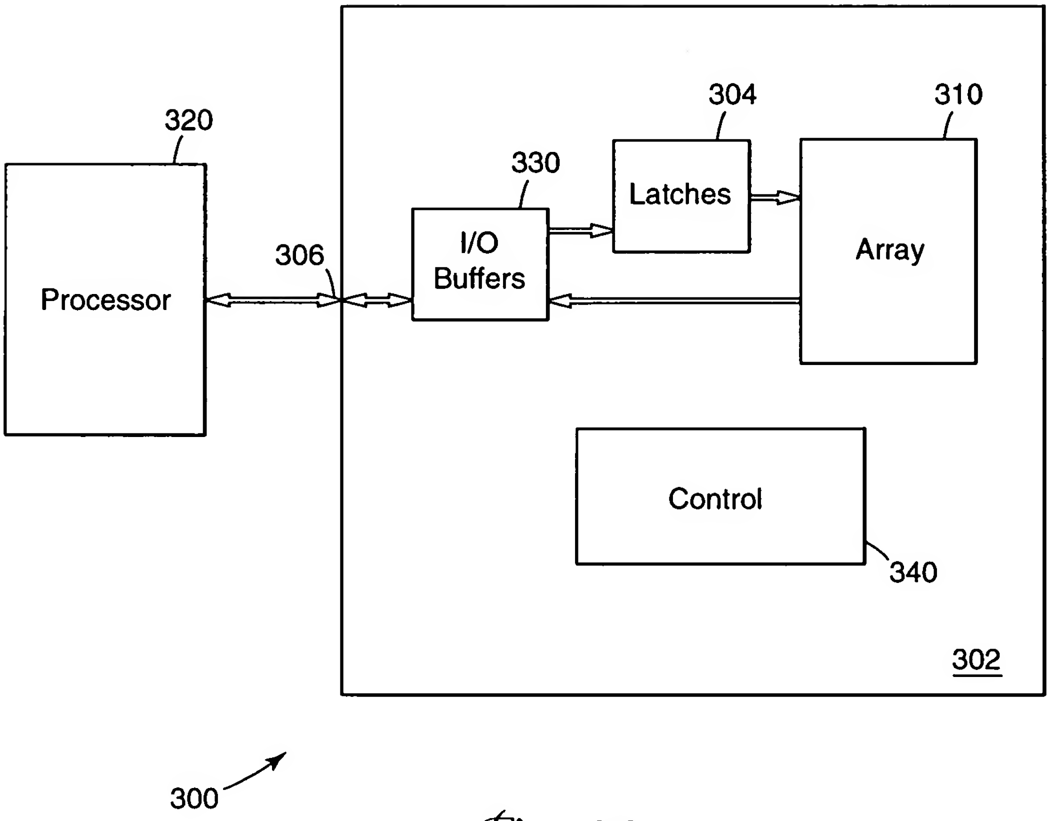

320
Processor
306
330
I/O
Buffers
304
Latches
310
Array
Control
340
302
300

Fig. 32

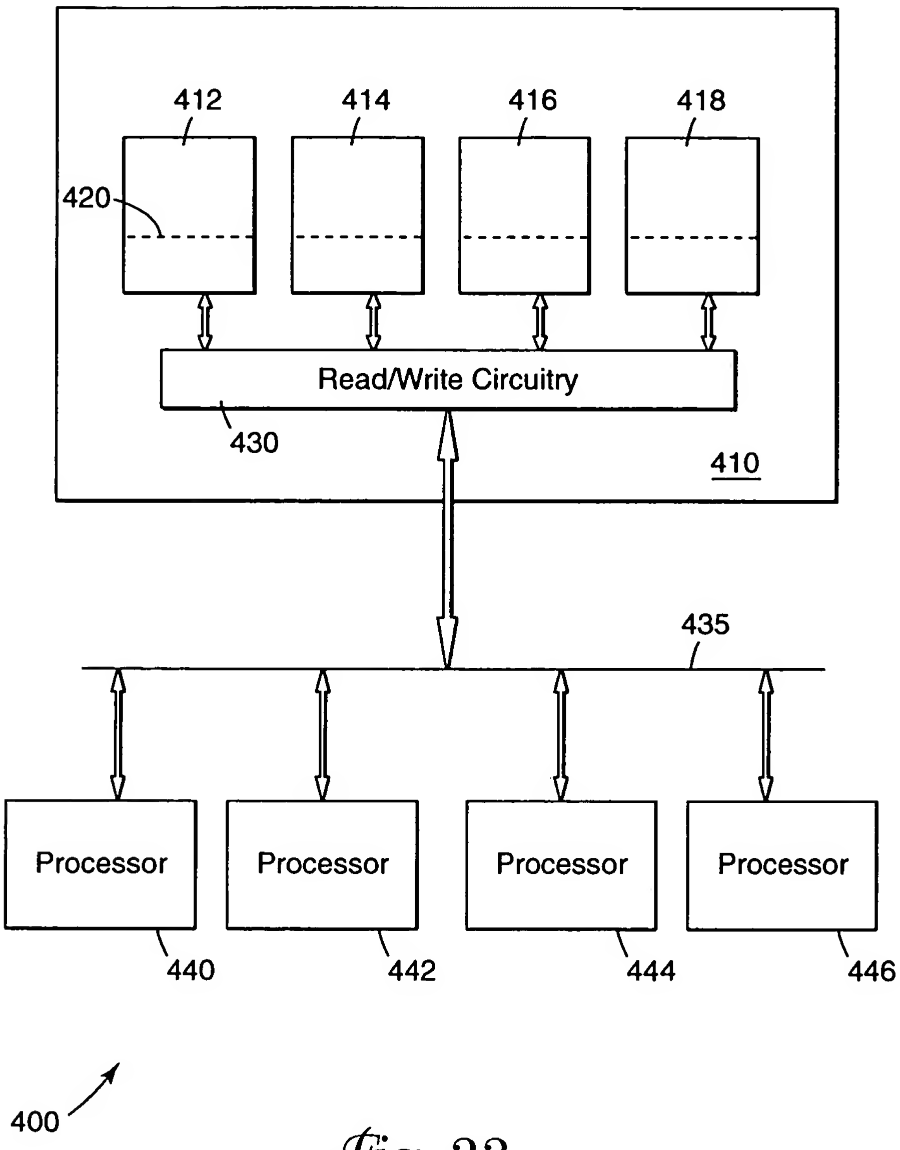

412
414
416
418
420
Read/Write Circuitry
430
410
435
Processor
Processor
Processor
Processor
440
442
444
446
400

Fig. 33

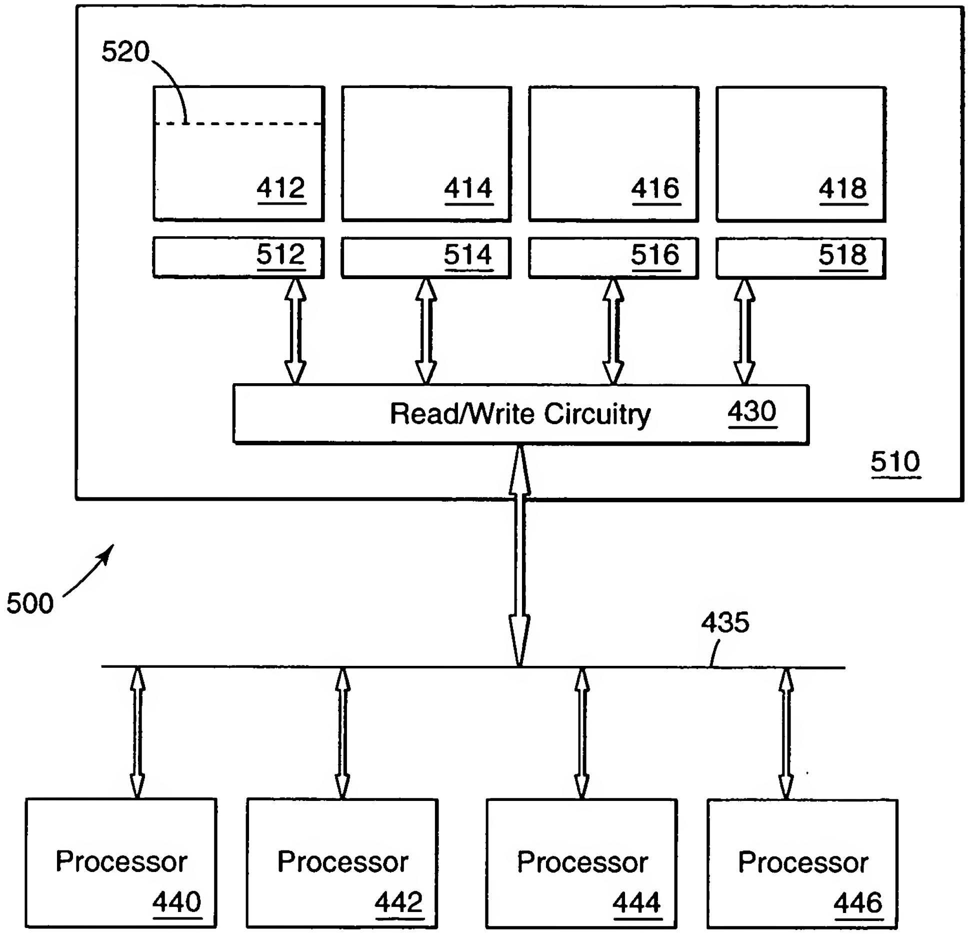

520
412
414
416
418
512
514
516
518
Read/Write Circuitry
430
510
500
435
Processor
440
Processor
442
Processor
444
Processor
446

Fig. 34